

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-141052

(43)Date of publication of application : 02.06.1995

(51)Int.Cl.

G06F 1/04  
G06F 1/04  
G06F 1/20  
G06F 1/08

(21)Application number : 06-132237

(71)Applicant : SEIKO EPSON CORP

(22)Date of filing : 14.06.1994

(72)Inventor : NAKAMURA AKIYOSHI

(30)Priority

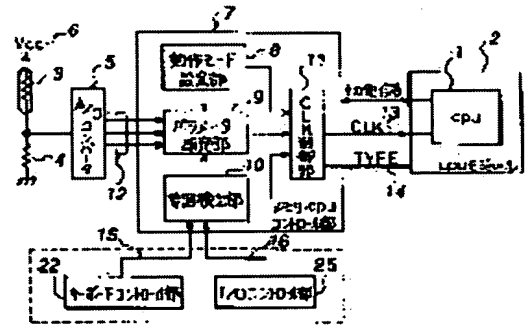
Priority number : 05235094 Priority date : 21.09.1993 Priority country : JP

## (54) INFORMATION PROCESSOR

(57)Abstract:

**PURPOSE:** To use a CPU provided with throughput to generate runaway by heat generation when a continuous operation is kept by controlling a temperature by changing a CLK frequency based on the maximum CLK frequency of the CPU, the CPU type and the ratio of operation time by frequencies at prescribed time.

**CONSTITUTION:** The CPU type is discriminated concerning the CLK frequency, kind of a CPU 1, kind of interrupt and CLK type. Next, an operation mode setting part 8 sets an operation mode concerning the temperature control and corresponding to that setting, control signals are outputted to a parameter setting part 9 and a CLK control part 11. Further the parameter setting part 9 outputs a control signal to the CLK control part 11 corresponding to the temperature provided by the temperature check, the set values of parameters and factors from a factor detection part 10 and the CLK control part 11 controls the temperature of the CPU 1 by controlling the frequency of a CLK 13. Thus, the radiation panel of the CPU is detached, and it can be made into a note personal computer.



## LEGAL STATUS

[Date of request for examination] 18.12.1998

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 3246186

[Date of registration] 02.11.2001

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(11)特許出願公開番号

特開平 7 - 1 4 1 0 5 2

(43)公開日 平成7年(1995)6月2日

(51) Int. Cl. <sup>6</sup>

G 0 6 F 1/04

識別記号

片内整理番号

F I

## 技術表示箇所

A

3 0 1 B

1/20

G 0 6 F

1/00

360 D

1/04

3 2 0 A

審査請求 未請求 請求項の数 2 1 0 L

(全 29 頁)

[最終頁に続く](#)

(21) 出願番号                      特願平6-132237

(22) 出願日 平成6年(1994)6月14日

(31) 優先權主張番号 特願平5-235094

(32) 優先日 平5(1993)9月21日

(33) 優先権主張国 日本 (JP)

(71)出願人 000002369

セイコーエプソン株式会社

東京都新宿区西新宿2丁目4番1号

(72)発明者 中村 明善

長野県諏訪市大和3丁目3番5号 セイコー

エプソン株式会社内

(74)代理人 弁理士 鈴木 喜三郎 (外1名)

(54) 【発明の名称】 情報処理装置

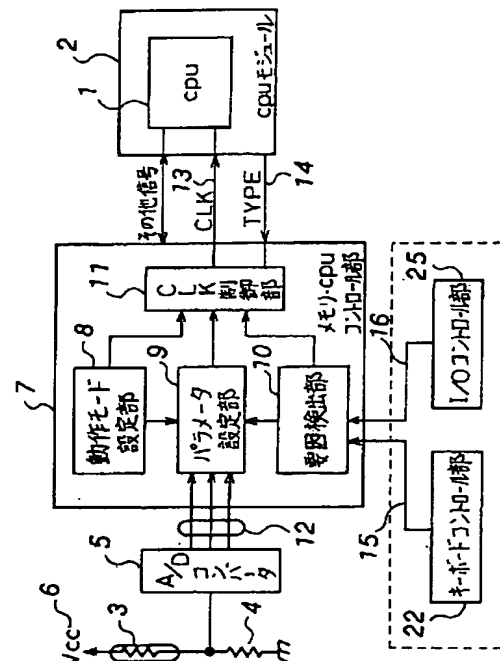
(57) 【要約】

【目的】 連続動作を続けると発熱によって熱暴走をする程の処理能力をもったCPUを温度制御を行なうことで情報処理装置に使用可能とすることを目的とする。

尚、そのCPUはグレードアップも可能である。

【構成】 CPUタイプの判別部、温度検出部、ユーザーによる動作モード設定部、及びシステムで決定される動作モードを持ち、ユーザーの情報処理装置に対する指示とソフトウェアの動作によってCLK周波数をリアルタイムに変更することで自動的に温度制御を行なう。

尚、CPUの温度制御手段はCLK周波数を変更することで行う手段に限定されない。



## 【特許請求の範囲】

【請求項 1】第 1 の CPU と、

前記第 1 の CPU よりも高い処理能力を有する第 2 の CPU を前記第 1 の CPU と交換可能にする手段と、  
前記第 1 の CPU 及び前記第 2 の CPU の最大 CLK 周波数を情報処理装置本体に通知する手段と、  
前記第 1 の CPU 及び前記第 2 の CPU タイプを前記情報処理装置本体に通知する手段とを有する CPU モジュールと、  
CPU の CLK 周波数を少なくとも 2 種類の周波数（第 1 の周波数と第 2 の周波数）にリアルタイムに変更する手段と、

第 1 の周波数に変更する手段を少なくとも入力部からの動作信号もしくは入力部からのデータをアクセスするための命令を含む特定の要因（以下 CLK 周波数変更要因と呼ぶ）によって行い、かつ CLK 周波数が第 1 の CLK 周波数に変更された後所定時間（以下 CLK 周波数保持時間と呼ぶ）後に前記特定の要因が検出されない場合、変更される前の第 2 の周波数に戻る機能を有する情報処理装置において、  
前記 2 種類の周波数ごとの動作時間（以下周波数別動作時間と呼ぶ）を検出する手段を有し、  
各々の CPU の最大 CLK 周波数と CPU タイプ及び所定時間における前記周波数別動作時間の割合に基づいて、前記第 1 の CLK 周波数を変更することを特徴とする情報処理装置。

【請求項 2】第 1 の CPU と、

前記第 1 の CPU よりも高い処理能力を有する第 2 の CPU を前記第 1 の CPU と交換可能にする手段と、  
前記第 1 の CPU 及び前記第 2 の CPU の最大 CLK 周波数を前記情報処理装置本体に通知する手段と、  
前記第 1 の CPU 及び前記第 2 の CPU タイプを前記情報処理装置本体に通知する手段とを有する CPU モジュールを持ち、  
前記 CPU は、前記 CPU 内部の CPU コアに対する CLK を外部信号（以下 CLK 停止信号とよぶ）によって止める手段を有し、  
更に前記 CLK 停止信号を一定周期で所定時間、前記 CPU に入力するモード（以下 CLK 停止モードとよぶ）を有し、

入力部からの動作信号もしくは入力部からのデータをアクセスするための命令を含む CLK 周波数変更要因が検出されたとき前記 CLK 停止モードを解除する（以下 CLK 動作モードとよぶ）手段と、CLK 周波数保持時間後に前記特定の要因が検出されない場合、前記 CLK 停止モードに戻る機能を有する情報処理装置において、  
前記 CLK 動作モード及び前記 CLK 停止モードごとの動作時間（周波数別動作時間）を検出する手段を有し、  
各々の CPU の最大 CLK 周波数と CPU タイプ及び所定時間における前記周波数別動作時間の割合に基づいて、

て、前記 CLK 停止信号を制御することを特徴とする情報処理装置。

【請求項 3】前記情報処理装置のケース内部の温度、又は CPU の内部温度、又は CPU の表面温度、又は CPU が実装されている基板上の温度を計測する温度計測手段とを有し、

前記周波数別動作時間と前記温度計測手段からの温度を示す信号により前記 CLK 周波数もしくは前記 CLK 停止信号を制御することを特徴とする請求項 1 ないし 2 記載の情報処理装置。

【請求項 4】CPU の CLK 周波数を少なくとも 2 種類の周波数（第 1 の周波数と第 2 の周波数）にリアルタイムに変更する手段と、

第 1 の周波数に変更する手段を少なくとも入力部からの動作信号もしくは入力部からのデータをアクセスするための命令を含む CLK 周波数変更要因によって行い、かつ CLK 周波数が第 1 の CLK 周波数に変更された後 CLK 周波数保持時間経過したあと前記 CLK 周波数変更要因が検出されない場合、変更される前の第 2 の周波数に戻る機能を有する情報処理装置において、

前記情報処理装置のケース内部の温度もしくは、CPU の表面温度もしくは CPU が実装されている基板上の温度を計測する温度計測手段を有し、  
前記温度計測手段からの温度を示す信号により前記第 1 の CLK 周波数を変更することを特徴とする情報処理装置。

【請求項 5】内部の CPU コアに対する CLK を CLK 停止信号によって止める手段を有する CPU と、

前記 CLK 停止信号を一定周期で所定時間、前記 CPU に入力する CLK 停止モードとを有し、

入力部からの動作信号もしくは入力部からのデータをアクセスするための命令を含む CLK 周波数変更要因が検出されたとき前記 CLK 停止モードを解除する CLK 動作モードと、CLK 周波数保持時間後に前記特定の要因が検出されない場合、前記 CLK 停止モードに戻る機能を有する情報処理装置において、

前記情報処理装置のケース内部の温度もしくは、CPU の表面温度もしくは CPU が実装されている基板上の温度を計測する温度計測手段を有し、

前記温度計測手段からの温度を示す信号により前記 CLK 停止信号を制御することを特徴とする情報処理装置。

【請求項 6】前記情報処理装置は、CLK 周波数を常に CPU の最大 CLK 周波数に固定するモード（以下ノーマルモードと呼ぶ）を更に有し、

キーボードのキーコンビネーションもしくは、キーコンビネーションか専用キーもしくはポインティングデバイスの指示によって現われるポップアップメニューによって、前記 CLK 周波数もしくは前記 CLK 停止信号を制御する手段を有するモードと、ノーマルモードとを切り換える手段を有することを特徴とする請求項 1、2、

4、5記載の情報処理装置。

【請求項 7】前記 CLK 周波数もしくは前記 CLK 停止信号を制御する手段に加えて、CPU の電源電圧も変化する手段を有することを特徴とする請求項 1、2、4、5 記載の情報処理装置。

【請求項 8】前記 CLK 周波数変更要因（本請求項では以下第 1 の CLK 周波数変更要因と呼ぶ）から選択された第 2 の CLK 周波数変更要因を有し、前記第 1 の周波数に変更する手段を第 2 の CLK 周波数変更要因によって行い、かつ CLK 周波数が第 1 の CLK 周波数に変更された後前記 CLK 周波数保持時間とは異なる第 2 の CLK 周波数保持時間が経過したあと前記第 2 の CLK 周波数変更要因が検出されない場合、第 2 の周波数に変更する手段（本請求項では以下第 2 の CLK 周波数変更手段と呼ぶ。尚、前述の第 1 の CLK 周波数変更要因に基づく CLK 周波数変更手段を第 1 の CLK 周波数変更手段と呼ぶ。）を有し、更に、最終的な CPU の CLK 周波数は、第 1 の CLK 周波数変更手段による周波数切り換え信号と第 2 の CLK 周波数変更手段による周波数切り換え信号の AND 信号によって切り換えられる手段を有することを特徴とする請求項 1 記載の情報処理装置。

【請求項 9】前記第 1 の CLK 周波数変更要因から選択された第 2 の CLK 周波数変更要因を有し、前記 CLK 停止モードに変更する手段を第 2 の CLK 周波数変更要因によって行い、かつ前記 CLK 停止モードに変更された後前記 CLK 周波数保持時間とは異なる第 2 の CLK 周波数保持時間が経過したあと前記第 2 の CLK 周波数変更要因が検出されない場合、CLK 動作モードに変更する第 2 の CLK 周波数変更手段を有し、更に、最終的な CLK 停止信号は、第 1 の CLK 周波数変更手段による周波数切り換え信号と第 2 の CLK 周波数変更手段による周波数切り換え信号の AND 信号によって切り換えられる手段を有することを特徴とする請求項 2 記載の情報処理装置。

【請求項 10】前記情報処理装置は、CLK 周波数もしくは前記 CLK 停止信号の周期とパルス幅を任意に設定できる手段を更に有し、所定の CLK 周波数以下に設定された場合、もしくは前記 CLK 停止信号の周期とパルス幅が所定の値に設定された場合前記第 2 の CLK 周波数変更手段を無効とする機能を有することを特徴とする請求項 8、9 記載の情報処理装置。

【請求項 11】CPU の CLK 周波数を少なくとも 2 種類の周波数（第 1 の周波数と第 2 の周波数）にリアルタイムに変更する手段と、第 1 の周波数に変更する手段を少なくとも入力部からの動作信号もしくは入力部からのデータをアクセスするための命令を含む CLK 周波数変更要因によって行い、かつ CLK 周波数が第 1 の CLK 周波数に変更された後 C

L 周波数保持時間経過したあと前記 CLK 周波数変更要因が検出されない場合、変更される前の第 2 の周波数に戻る機能を有する情報処理装置において、前記 CLK 周波数変更要因ごとに、要因検出後の周波数を設定する機能を有することを特徴とする情報処理装置。

【請求項 12】内部の CPU コアに対する CLK を CLK 停止信号によって止める手段を有する CPU と、前記 CLK 停止信号を一定周期で所定時間、前記 CPU に入力する CLK 停止モードとを有し、入力部からの動作信号もしくは入力部からのデータをアクセスするための命令を含む CLK 周波数変更要因が検出されたとき前記 CLK 停止モードを解除する CLK 動作モードと、CLK 周波数保持時間後に前記特定の要因が検出されない場合、前記 CLK 停止モードに戻る機能を有する情報処理装置において、前記 CLK 周波数変更要因ごとに、要因検出後の CLK 停止信号の周期とパルス幅を設定する機能を有することを特徴とする情報処理装置。

【請求項 13】前記 CLK 周波数変更要因によって、CLK 周波数もしくは CLK 停止信号の周期とパルス幅ばかりでなく CPU の電源電圧も変化する手段と、前記 CLK 周波数変更要因ごとに、要因検出後の電圧も設定できる手段を有することを特徴とする請求項 11、12 記載の情報処理装置。

【請求項 14】前記 CPU のタイプが、CLK 信号を CPU 内部で PLL を使用して逡倍化して使用するタイプの場合、前記 CPU 内部に、前記 PLL を介して入力される第 1 の CLK 信号と、前記 PLL を介さない第 2 の CLK 信号とを有し、前記 CPU 内部に前記第 1 の CLK 信号と、前記第 2 の CLK 信号とを選択するセレクタを有することを特徴とする請求項 1 ないし 2 記載の情報処理装置。

【請求項 15】前記 CPU のタイプや CPU の最大 CLK 周波数によって、前記 CLK 周波数保持時間を変更する手段を有することを特徴とする請求項 1 記載の情報処理装置。

【請求項 16】前記情報処理装置のケース内部の温度、又は CPU の内部温度、又は CPU の表面温度、又は CPU が実装されている基板上の温度の各温度によって、空冷ファンの回転数を制御する手段を有することを特徴とする請求項 3、4、5 記載の情報処理装置。

【請求項 17】情報処理装置本体と取り外し可能な CPU モジュールに実装されている CPU の温度を検出する手段を有する情報処理装置において、前記温度検出手段が温度検出素子と前記温度検出素子と直列に接続された抵抗からなることを特徴とする情報処理装置。

【請求項 18】CPU の温度を検出する手段と、前記温

度検出手段により、前記CLK周波数もしくは前記CLK停止信号を制御することで前記CPUの処理速度を低下させる手段を有する情報処理装置において、前記CPUの処理速度が低下したことを現す表示手段を有することを特徴とする情報処理装置。

【請求項19】内部のCPUコアに対するCLKをCLK停止信号によって止める手段を有するCPUと、前記CLK停止信号を一定周期で所定時間、前記CPUに入力するCLK停止モードとを有する情報処理装置において、前記CLK停止信号が入力されると同時に、前記CPUへ出力するCLKの周波数も変化させることを特徴とする情報処理装置。

【請求項20】CPUのCLK周波数を少なくとも2種類の周波数（第1の周波数と第2の周波数）にリアルタイムに変更する手段と、第1の周波数に変更する手段を少なくとも入力部からの動作信号もしくは入力部からのデータをアクセスするための命令を含むCLK周波数変更要因によって行い、かつCLK周波数が第1のCLK周波数に変更された後CLK周波数保持時間経過したあと前記CLK周波数変更要因が検出されない場合、変更される前の第2の周波数に戻る第1のモードと、

CLK周波数を常に第1の周波数に固定する第2のモードを有する情報処理装置において、前記情報処理装置のケース内部の温度もしくは、CPUの表面温度もしくはCPUが実装されている基板上の温度を計測する温度計測手段と、前記温度計測手段からの温度を示す信号により前記CLK周波数を変更する手段を有し、更に、前記温度計測手段からの温度データにより前記第1のモードと第2のモードを切り換える手段を有することを特徴とする情報処理装置。

【請求項21】内部のCPUコアに対するCLKをCLK停止信号によって止める手段を有するCPUと、前記CLK停止信号を一定周期で所定時間、前記CPUに入力するCLK停止モードとを有し、入力部からの動作信号もしくは入力部からのデータをアクセスするための命令を含むCLK周波数変更要因が検出されたとき前記CLK停止モードを解除するCLK動作モードと、CLK周波数保持時間後に前記特定の要因が検出されない場合、前記CLK停止モードに戻る第1のモードと、CLK周波数変更要因に影響されずCLK停止信号を制御する第2のモードとを有する情報処理装置において、前記情報処理装置のケース内部の温度もしくは、CPUの表面温度もしくはCPUが実装されている基板上の温度を計測する温度計測手段と、前記温度計測手段からの温度を示す信号により前記CLK停止信号を制御する手段と、

前記温度計測手段からの温度データにより前記第1のモードと第2のモードを切り換える手段と、を有することを特徴とする情報処理装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、CLK周波数を変更可能な情報処理装置に関する。

【0002】

【従来の技術】従来、携帯可能なパーソナルコンピュータ（以下携帯型PCという）のCPUに対する発熱対策としては強制的に熱を奪う方法と、消費電力を減らすことにより発熱を押さえる方法が一般的であった。例えば次のような方法があった。

【0003】強制的に熱を奪う方法としては、

- （1）強制空冷用の扇風機（以下ファンとする）を用いる
- （2）放熱板をCPUに装着する事により、熱を分散させて、全体の発熱を均一化させる
- （3）装置のケースに、空気を取り入れる穴を空け、放熱する。

【0004】消費電力を減らす事により発熱を押さえる方法としては、

- （1）ICのデザインルールを変更しチップサイズを変更することで消費電力を減らす
- （2）情報処理装置の電源電圧を落として、消費電力を減らす。これは、現在最も一般的な電源電圧5Vを、例えば3.3Vなどに落とし、消費電力を減らす方法など様々な方法を行ってきた。そして、これらの対策は基本的に設計上の最大電力時の発熱に対して、機器の安全性を保証するために行なっている。

【0005】ところが現在CPUの処理能力が向上しそれに伴って消費電力が多くなってきたために、前記各方法をもってしても携帯型PCに対する熱対策に限界が見えてきた。まず強制的に熱を奪う方法としてあげた前記対策は、携帯型PCに対してはもともと小さくするということと、空冷ファンをつけなければならない又は放熱板をつけなければならないということが矛盾するので適切な放熱効果を得ることは非常に難しい。他に、屋内でACアダプタを使用する時のみ空冷ファンのついたユニットを取り付けCLK周波数をあげて使用し、屋外で電池動作で使用する場合は空冷ファンのついたユニットを取りはずしCLK周波数を落とすものもあるが、携帯型PCにつく風力に限界があることと、電池動作時にCPUパワーをフルにいかせないのには大きな問題がある。

【0006】一方ケースが比較的大きいデスクトップ型PCにおいても、放熱板の大きさ、空冷ファンによる放熱効果に限界があり熱対策をおこなうのが徐々に難しくなっている。又、空冷ファンによる騒音がオフィスの環境問題の点で問題視されつつあり今以上の強制空冷手段を設けるのが難しくなっている。

【0007】そこで現在ではいかに消費電力を減らす事により発熱を押さえられるかということが非常に重要になってきている。ところが、消費電力を減らす前述の方法でも、デザインルールの進歩、電源電圧の低電圧化に対してあまりにもCPUの処理能力の向上に伴う消費電力の増加が多いため限界が見えてきている。

【0008】以上から、従来の延長線上では熱問題のためにCPUの処理能力向上に限界がでてきた事がわかる。そこで、特に携帯型PCにおいて更に処理能力の高い(従って消費電力の多い)CPUを使用するために従来は大別して4つの方法があった。以下に各方法について従来例をもちいながら説明する。

【0009】(1)、温度センサーをもちCPUが一定の温度になったらCLK周波数を落としたりHOLD信号、キャッシュ制御等を行う。

【0010】この例としては、特開平2-181252号公報等がある。特開平2-181252号公報では、CPUのクロック入力装置に供給するクロック信号の周波数を、周囲温度等により、バスサイクルの実行時間を制御することを特徴としている。詳しく説明すると、周囲温度を検出するセンサーを設けて、前記センサーが設定した温度以上になるとCPUのクロックを遅くする。この特許の目的自体は、温度に応じてバスサイクルの実行時間を制御することで最適なパフォーマンスを得るところにあるが機能自体は結果的に温度制御になっている。

【0011】また別の対策としては、温度センサーにより設定された阻止破壊温度以上になるとシステム全体を停止する方法が考えられる。この例としては、特開平3-11409号公報がある。特開平3-11409号公報では、2つのセンサー(サーマルスイッチ)を設けて、検出温度に応じてクロックの発振手段が出力するクロック信号の周波数を変化させる手段と、阻止破壊温度以上になるとシステムの動作を停止させる手段とを備えている。以上の方法では、センサーがあらかじめ決められた一定温度になると強制的にCLK周波数がおとされるのでそれ以降の処理が実質的に遅くなってしまいうという欠点がある。

【0012】温度センサーによって、CPUのクロック信号ではなくHOLD信号、キャッシュ信号を制御することでCPUの温度を落とす実施例として筆者を含む共同出願の特願平4-310540号がある。

【0013】(2)、CPUが動作する場合だけ高いCLK周波数にして、動作しない場合には低い周波数とする。ただし、これは温度に対するものではなくパワーセーブの為の機能である。

【0014】本方法は直接的には、CPUの温度制御を目的としているのではなくパワーセーブを目的としている機能である。しかし、強制的に熱を奪う方法にも従来の方法で消費電力を落とすのにも限界が見えた今、パワ

ーセーブに限らずCPUが高速動作を必要としているときだけ高速動作し、キーボード入力待ちの時などCPUの処理を必要としない時には低速動作することのできる機能は必要不可欠である。処理量が一定であれば、CPUの処理能力があがればあがるほど処理時間は短くて済むはずでありそれに従えばCPUの処理能力があがってもパワー及び温度は一定に保つことができるはずである。

【0015】この例としては、筆者出願の特開平4-354009号公報、特願平3-164657号がある。特開平4-354009号公報ではキーボード、マウス、FDD等のアクセス信号を受けてからある特定の時間だけ高速動作を行い、何もアクセスがおこっていない時には低速動作をおこなうというものである。特願平3-164657号では、低速動作時には同時に電圧も落とすというものである。

【0016】他にメモリに対するアドレスを監視することで現在CPUが実行中なのか、単純ループで何らかの割り込みを待っているのかを判別しCPUが実行中でないときにはCLK周波数を落とす方法も考えられている。又、アプリケーション側からプログラムがアイドル状況であるということをシステムに対して通知しCLK周波数を落とすという方法もある。上記どの方法によっても、プログラムを限定せずCPUが実行中であるかどうかを完全に判別するのは難しい。従って当り前の事であるが、電池寿命の問題、熱の問題がなければ常に最高の周波数でCPUを動作させることがパフォーマンスを損なわない1番の方法である。

【0017】(3)、温度センサーを持たず、ある一定時間のCLK周波数の高周波数と低周波数の割合から温度上昇カーブを推定して、あらかじめ決められた温度限界に達したところでCLK周波数を低くする。

【0018】特定のCPUに対するCLK周波数と動作時間がわかれば概略の温度上昇カーブが推察できる。これを利用して、ある特定の時間(以下温度カーブ検出時間とよぶ。ここでは10分とする)を決めその時間内の高速と低速の動作の割合をたとえば60%と決めた場合、6分は高速動作を行い、後の4分は低速動作を行うことでCPUの温度上昇を防ぐ方法も知られている。本方法と上記(2)で示した方法を組み合わせた場合、何もアクセスがおこっていない時には低速動作をおこなうので、高速動作時間の合計値が6分に達するまで見かけ上パフォーマンスを落とすことなく温度制御を行うことができる。もちろん10分を越える連続動作が生じた場合CPUは放熱上の観点から6分経過した後4分間は低速動作となるので平均では60%~70%程度の処理能力しか発揮できない。しかし、MS-DOS上で動作するワープロ等のプログラムでは連続動作がほとんど生じることはないので実際に温度限界によって低速動作になることは少ない。問題が生じるのはWINDOWSのよ

うに組み込まれたデバイスドライバが一定間隔でデバイスをチェックするプログラムであり、このようなプログラムでは常に連続動作が発生することになり、実際CPUパワーが必要なときに温度限界によって低速動作になってしまう。(MS-DOS、WINDOWSはマイクロソフト社の商標である)

(4) . 温度センサーからの出力によりCPUに対し割り込みを発生させることで、ソフト的にCPUの熱保護を行う。

【0019】実施例として筆者を含む共同出願の特願平4-310540号がある。温度センサーからの出力によりCPUに対し割り込みを発生させることで、ソフト的に温度制御を行うものでありこの時用いられる割り込みは、互換性を損なわず透過的に発行できるシステムマネージメントインタラプト(以下SMIという)という割り込みであり、アプリケーションに左右されない優先順位の高い割り込みを発生することができる。SMIがある場合ハードウェアで完全に制御しなくてもよいのでフレキシブルな温度制御が可能となる。

【0020】さて以上はCPUがあらかじめ実装されているものについて主に述べてきたが、近年CPUが本体と脱着可能な構造を持ちCPU自体をグレードアップすることができるデスクトップPC、携帯型PCが増えてきている。ここでグレードアップされるCPUがまだ現実にはない場合では、従来は温度保証が難しかった。それは実際に温度上昇検査をしてみなければCPU規格を満たすかどうか分からないからである。ところが、資源活用の面、コストの面からできるだけ長くPCを使いたいという欲求と同時に最新のCPUパワーを欲しいという欲求も出てきている。そこで、CPUがまだ現実にはない場合でも温度的になんらかの保証手段をPC内にあらかじめ備えておくことが求められている。このような場合に対応する為の従来例を以下に示す。

【0021】(1) . CPUソケット部などに温度センサーを設けて温度制御する。実施例として前述の特願平4-310540号がある。図10においてCPUが装着されるICソケット1001のまん中に温度センサー1002を設けることでCPUがグレードアップされても温度制御することを可能にしている。

【0022】(2) . 交換可能なCPUモジュールから周波数、CPUタイプなどの情報を受けてグレードアップされたCPUに対してそれぞれに最適なパワーセーブ及び温度制御を実現する。実施例として筆者出願の特願平5-154786号がある。本願では、PQFP(PLASTIC QUAD FLATPACK PACKAGE)、PGA(PIN GRID ARRAY)などの外形をもつCPUをプリント基板上に実装して、そのプリント基板上に設けられた周波数、CPUタイプなどを設定するハンダジャーバーを変更する事によりCPUごとに異なった情報を本体に通知する。本体はそれを受

けて各CPUに対して適切なパワーセーブと温度制御を行なう。

#### 【0023】

【発明が解決しようとする課題】これまで述べてきたように、特に携帯型PCにおいてパワーセーブ及び温度制御についてさまざまな方法が考えられてきた。またそれはCPUがグレードアップされることまで考慮され実現されてきた。ところが従来の方法では、更に処理能力の高い(従って消費電力の多い)CPUを使用することのできるシステムが構築されたとはいえない。それは、前述のどの方法によってもCPUの熱保護とその本来の目的である、更に処理能力の高いCPUを使用することのバランスが悪く実現性に乏しいところにある。まず、CPUの発熱を温度センサーによって検出し、そのレベルによってCLK周波数を下げたり、HOLD挿入時間を変えたり、LCD等のバックライトを消すなどして本体内部の温度上昇を防ぐような機能では、CPUの発熱を温度センサーによって検出した後には実質的にはCPUのパワーをユーザーは使用できなくなってしまう。これでは、CPUの処理能力が高くなり発熱量が大きくなっていくと電源を入れてから数10分しかPCをフルパワーで使用できないという本末転倒の状況さえ生まれかねない。

【0024】そこで考えられたのが、前述した温度センサーを持たず、ある一定時間のCLK周波数の高周波数と低周波数の割合から温度上昇カーブを推定して、あらかじめ決められた温度限界に達したところでCLK周波数を低くする方法である。これは、動作中のプログラムがCPUパワーを必要としているときのみCPUのCLK周波数をあげることを前提に、ある一定時間ごとにCLK周波数を細かく制御するものなので、CPUの発熱を温度センサーによって検出するよりもより長い期間CPUパワーを得ることができる。前述したように、この方法はMS-DOS上で動作するワープロ等のような通常のアプリケーションプログラムの動作ではほとんど連続動作が生じることはないことを前提としている。本方法はセンサーを必要としないが問題がいくつかある。

【0025】(1) . WINDOWSでは温度限界によって低速動作になってしまう。前述したようにWINDOWSのように組み込まれたデバイスドライバが一定間隔でデバイスをチェックするプログラムの場合、結果的に常に連続動作が発生することになり、実際にCPUパワーが必要なときに温度限界によって低速動作になってしまう。

【0026】(2) . 温度センサーを持たないため環境温度の変化に対応できない。CPUの温度やCPUの周囲温度をはかるセンサーからの出力によらず、動作時間とCLK周波数のみで制御するので、機器の保証上からもマージンが必要なためどうしても最悪環境下での設定になりCPUパワーをフルにいかしきれない。

【0027】(3)、温度カーブ検出時間を越える長時間の計算を必要とする場合、CPUのパワーが制限されてしまう。前述したように、温度カーブ検出時間を10分とし温度制限を60%と設定すると、10分を越える連続動作が生じた場合CPUは放熱上の観点から6分経過した後4分間は低速動作となるので平均では60%~70%程度の処理能力しか発揮できない。

【0028】(4)、CPUの温度上昇はCLK周波数だけで決まらない。CPUの温度上昇は、CLK周波数だけでなく、発生するサイクル、内部キャッシュなど使用されている機能によって異なる。又、CPU内部の構成やCPU内部のトランジスタ数、最小線幅(デザインルール)によっても異なる。したがって本方式では、マージンをもって設計するとCPUのパワーを落とす方向になってしまう。

【0029】(5)、CPUがグレードアップ可能な情報機器のように数種類のCPUに対応が困難である。CPUによって動作時間による温度上昇カーブが異なるため、CLK周波数のみでは完全には温度上昇カーブをシミュレーションできない。従ってCPUを何段階にもグレードアップ可能な情報機器の場合適さない。本方法に限らずCPUがグレードアップ可能な情報機器に対応する発熱対策自体が従来存在していなかった。

【0030】本発明は上記問題点を解決することを目的としており、連続動作を続けると発熱によって熱暴走してしまうくらいの処理能力をもったCPUを、その処理能力を損なうことなくまた発熱による熱暴走を起こすことのないように温度制御することによって使用可能にする情報処理装置を提供することを目的とする。結果的に、本来大型の放熱板を有するCPUを放熱板を取ることで携帯型のノート型パーソナルコンピュータ(以下ノートパソコン)に装着することを可能とすることを目的としている。

【0031】また特に携帯型のノートパソコン等において、グレードアップされるCPUに応じて最適な発熱コントロールを行うことを可能とする情報処理装置を提供することも目的とする。

#### 【0032】

【課題を解決するための手段】本発明の情報処理装置は、第1のCPUと前記第1のCPUよりも高い処理能力を有する第2のCPUを前記第1のCPUと交換可能にする手段と、前記第1のCPU及び前記第2のCPUの最大CLK周波数を前記情報処理装置本体に通知する手段と、前記第1のCPU及び前記第2のCPUタイプを前記情報処理装置本体に通知する手段とを有するCPUモジュールと、CPUのCLK周波数を少なくとも2種類の周波数(第1の周波数と第2の周波数)にリアルタイムに変更する手段と、第1の周波数に変更する手段を少なくとも入力部からの動作信号もしくは入力部からのデータをアクセスするための命令を含む特定の要因

(以下CLK周波数変更要因と呼ぶ)によって行い、かつCLK周波数が第1のCLK周波数に変更された後所定時間(以下CLK周波数保持時間と呼ぶ)後に前記特定の要因が検出されない場合、変更される前の第2の周波数に戻る機能を有する情報処理装置において、前記2種類の周波数ごとの動作時間(以下周波数別動作時間と呼ぶ)を検出する手段を有し、各々のCPUの最大CLK周波数とCPUタイプ及び所定時間における前記周波数別動作時間の割合に基づいて、前記CLK周波数を変更する事の特徴とする。

【0033】また、本発明の他の情報処理装置は、第1のCPUと前記第1のCPUよりも高い処理能力を有する第2のCPUを前記第1のCPUと交換可能にする手段と、前記第1のCPU及び前記第2のCPUの最大CLK周波数を前記情報処理装置本体に通知する手段と、前記第1のCPU及び前記第2のCPUタイプを前記情報処理装置本体に通知する手段とを有するCPUモジュールを持ち、前記CPUは、前記CPU内部のCPUコアに対するCLKを外部信号(以下CLK停止信号とよぶ)によって止める手段を有し、更に前記CLK停止信号を一定周期で所定時間、前記CPUに入力するモード(以下CLK停止モードとよぶ)を有し、入力部からの動作信号もしくは入力部からのデータをアクセスするための命令を含むCLK周波数変更要因が検出されたとき前記CLK停止モードを解除する(以下CLK動作モードとよぶ)手段と、CLK周波数保持時間後に前記特定の要因が検出されない場合、前記CLK停止モードに戻る機能を有する情報処理装置において、前記CLK動作モード及び前記CLK停止モードごとの動作時間(周波数別動作時間)を検出する手段を有し、各々のCPUの最大CLK周波数とCPUタイプ及び所定時間における前記周波数別動作時間の割合に基づいて、前記CLK停止信号を制御する事の特徴とする。

【0034】また、前記情報処理装置のケース内部の温度、又はCPUの内部温度、又はCPUの表面温度、又はCPUが実装されている基板上の温度を計測する温度計測手段とを有し、前記周波数別動作時間と前記温度計測手段からの温度を示す信号により前記CLK周波数もしくは前記CLK停止信号を制御することを特徴とする。

【0035】また、本発明の他の情報処理装置は、CPUのCLK周波数を少なくとも2種類の周波数(第1の周波数と第2の周波数)にリアルタイムに変更する手段と、第1の周波数に変更する手段を少なくとも入力部からの動作信号もしくは入力部からのデータをアクセスするための命令を含むCLK周波数変更要因によって行い、かつCLK周波数が第1のCLK周波数に変更された後CLK周波数保持時間経過したあと前記CLK周波数変更要因が検出されない場合、変更される前の第2の周波数に戻る機能を有する情報処理装置において、前記

10

20

30

40

50



2種類の周波数ごとの動作時間を検出する手段と、前記情報処理装置のケース内部の温度もしくは、CPUの表面温度もしくはCPUが実装されている基板上の温度を計測する温度計測手段とを有し、前記周波数別動作時間と前記温度計測手段からの温度を示す信号により前記CLK周波数を変更することを特徴とする。

【0036】また本発明の他の情報処理装置は、内部のCPUコアに対するCLKをCLK停止信号によって止める手段を有するCPUと、前記CLK停止信号を一定周期で所定時間、前記CPUに入力するCLK停止モードとを有し、入力部からの動作信号もしくは入力部からのデータをアクセスするための命令を含むCLK周波数変更要因が検出されたとき前記CLK停止モードを解除するCLK動作モードと、CLK周波数保持時間後に前記特定の要因が検出されない場合、前記CLK停止モードに戻る機能を有する情報処理装置において、前記CLK動作モード及び前記CLK停止モードごとの動作時間（周波数別動作時間）を検出する手段を有し、前記情報処理装置のケース内部の温度もしくは、CPUの表面温度もしくはCPUが実装されている基板上の温度を計測する温度計測手段とを有し、前記周波数別動作時間と前記温度計測手段からの温度を示す信号により前記CLK停止信号を制御することを特徴とする。

【0037】更に、CLK周波数を常にCPUの最大CLK周波数に固定するモード（以下ノーマルモードと呼ぶ）を有し、キーボードのキーコンビネーションもしくは、キーコンビネーションか専用キーを叩くことによって現われるポップアップメニューによって、前記CLK周波数もしくは前記CLK停止信号を制御する手段を有するモードと、ノーマルモードとを切り換える手段を有する事を特徴とする。

【0038】また、前記CLK周波数もしくは前記CLK停止信号を制御する手段に加えて、CPUの電源電圧も変化させる手段を有することを特徴とする。

【0039】また、前記CLK周波数変更要因（本請求項では以下第1のCLK周波数変更要因と呼ぶ）から選択された第2のCLK周波数変更要因を有し、前記第1の周波数に変更する手段を第2のCLK周波数変更要因によって行い、かつCLK周波数が第1のCLK周波数に変更された後前記CLK周波数保持時間とは異なる第2のCLK周波数保持時間が経過したあと前記第2のCLK周波数変更要因が検出されない場合、第2の周波数に変更する手段（本請求項では以下第2のCLK周波数変更手段と呼ぶ。尚、前述の第1のCLK周波数変更要因に基づくCLK周波数変更手段を第1のCLK周波数変更手段と呼ぶ。）を有し、更に、最終的なCPUのCLK周波数は、第1のCLK周波数変更手段による周波数切り換え信号と第2のCLK周波数変更手段による周波数切り換え信号のAND信号によって切り換えられる手段を有することを特徴とする。

【0040】また、前記第1のCLK周波数変更要因から選択された第2のCLK周波数変更要因を有し、前記CLK停止モードに変更する手段を第2のCLK周波数変更要因によって行い、かつ前記CLK停止モードに変更された後前記CLK周波数保持時間とは異なる第2のCLK周波数保持時間が経過したあと前記第2のCLK周波数変更要因が検出されない場合、CLK動作モードに変更する第2のCLK周波数変更手段を有し、更に、最終的なCLK停止信号は、第1のCLK周波数変更手段による周波数切り換え信号と第2のCLK周波数変更手段による周波数切り換え信号のAND信号によって切り換えられる手段を有することを特徴とする。

【0041】また、CLK周波数もしくは前記CLK停止信号の周期とパルス幅を任意に設定できる手段を更に有し、所定のCLK周波数以下に設定された場合、もしくは前記CLK停止信号の周期とパルス幅が所定の値に設定された場合前記第2のCLK周波数変更手段を無効とする機能を有することを特徴とする。

【0042】更に、本発明の他の情報処理装置は、CPUのCLK周波数を少なくとも2種類の周波数（第1の周波数と第2の周波数）にリアルタイムに変更する手段と、第1の周波数に変更する手段を少なくとも入力部からの動作信号もしくは入力部からのデータをアクセスするための命令を含むCLK周波数変更要因によって行い、かつCLK周波数が第1のCLK周波数に変更された後CLK周波数保持時間経過したあと前記CLK周波数変更要因が検出されない場合、変更される前の第2の周波数に戻る機能を有する情報処理装置において、前記CLK周波数変更要因ごとに、要因検出後の周波数を設定する機能を有することを特徴とする。

【0043】また、本発明の他の情報処理装置は、内部のCPUコアに対するCLKをCLK停止信号によって止める手段を有するCPUと、前記CLK停止信号を一定周期で所定時間、前記CPUに入力するCLK停止モードとを有し、入力部からの動作信号もしくは入力部からのデータをアクセスするための命令を含むCLK周波数変更要因が検出されたとき前記CLK停止モードを解除するCLK動作モードと、CLK周波数保持時間後に前記特定の要因が検出されない場合、前記CLK停止モードに戻る機能を有する情報処理装置において、前記CLK周波数変更要因ごとに、要因検出後のCLK停止信号の周期とパルス幅を設定する機能を有することを特徴とする。

【0044】更に、前記CLK周波数変更要因によって、CLK周波数もしくはCLK停止信号の周期とパルス幅ばかりでなくCPUの電源電圧も変化させる手段と、前記CLK周波数変更要因ごとに、要因検出後の電圧も設定できる手段を有することを特徴とする。

【0045】また、前記CPUのタイプが、CLK信号をCPU内部でPLLを使用して通信化して使用するタ

10

20

30

40

50

イブの場合、前記CPU内部に、前記PLLを介して入力される第1のCLK信号と、前記PLLを介さない第2のCLK信号とを有し、前記CPU内部に前記第1のCLK信号と、前記第2のCLK信号とを選択するセレクタを有する事を特徴とする。

【0046】また、前記CPUのタイプやCPUの最大CLK周波数によって、前記CLK周波数保持時間を変更する手段を有する事を特徴とする。

【0047】また、前記情報処理装置のケース内部の温度、又はCPUの内部温度、又はCPUの表面温度、又はCPUが実装されている基板上の温度の各温度によって、空冷ファンの回転数を制御する手段を有する事を特徴とする。

【0048】また、本発明の他の情報処理装置は、情報処理装置本体と取り外し可能なCPUモジュールに実装されているCPUの温度を検出する手段を有する情報処理装置において、前記温度検出手段が温度検出素子と前記温度検出素子と直列に接続された抵抗からなることを特徴とする。

【0049】また、本発明の他の情報処理装置は、CPUの温度を検出する手段と、前記温度検出手段により、前記CLK周波数もしくは前記CLK停止信号を制御することで前記CPUの処理速度を低下させる手段を有する情報処理装置において、前記CPUの処理速度が低下したことを現す表示手段を有することを特徴とする。

【0050】また、本発明の他の情報処理装置は、内部のCPUコアに対するCLKをCLK停止信号によって止める手段を有するCPUと、前記CLK停止信号を一定周期で所定時間、前記CPUに入力するCLK停止モードとを有する情報処理装置において、前記CLK停止信号が入力されると同時に、前記CPUへ出力するCLKの周波数も変化させることを特徴とする。

【0051】また、本発明の他の情報処理装置は、CPUのCLK周波数を少なくとも2種類の周波数(第1の周波数と第2の周波数)にリアルタイムに変更する手段と、第1の周波数に変更する手段を少なくとも入力部からの動作信号もしくは入力部からのデータをアクセスするための命令を含むCLK周波数変更要因によって行い、かつCLK周波数が第1のCLK周波数に変更された後CLK周波数保持時間経過したあと前記CLK周波数変更要因が検出されない場合、変更される前の第2の周波数に戻る第1のモードと、CLK周波数を常に第1の周波数に固定する第2のモードを有する情報処理装置において、前記情報処理装置のケース内部の温度もしくは、CPUの表面温度もしくはCPUが実装されている基板上の温度を計測する温度計測手段と、前記温度計測手段からの温度を示す信号により前記CLK周波数を変更する手段を有し、更に、前記温度計測手段からの温度データにより前記第1のモードと第2のモードを切り換える手段を有することを特徴とする。

【0052】また、本発明の他の情報処理装置は、内部のCPUコアに対するCLKをCLK停止信号によって止める手段を有するCPUと、前記CLK停止信号を一定周期で所定時間、前記CPUに入力するCLK停止モードとを有し、入力部からの動作信号もしくは入力部からのデータをアクセスするための命令を含むCLK周波数変更要因が検出されたとき前記CLK停止モードを解除するCLK動作モードと、CLK周波数保持時間後に前記特定の要因が検出されない場合、前記CLK停止モードに戻る第1のモードと、CLK周波数変更要因に影響されずCLK停止信号を制御する第2のモードとを有する情報処理装置において、前記情報処理装置のケース内部の温度もしくは、CPUの表面温度もしくはCPUが実装されている基板上の温度を計測する温度計測手段と、前記温度計測手段からの温度を示す信号により前記CLK停止信号を制御する手段と、前記温度計測手段からの温度データにより前記第1のモードと第2のモードを切り換える手段と、を有することを特徴とする。

【0053】

【実施例】以下、本発明について3つの実施例に基づき詳細に説明する。実施例1では全体的な構想と説明を行い、実施例2では実施例1を部分的に具体化した例を示す。又、実施例3では実施例1の一部の他の例を示す。

【0054】<実施例1>図3に本発明の実施例としてノートパソコンのブロック図を示す。前記ノートパソコンはCPUモジュール2、メインメモリ27やBIOS ROM24などをコントロールするメモリ・CPUコントロール部7、LCD28及びCRT29を制御するVIDEO回路部30、キーボード31を制御するキーボードコントロール部22、FDD32やICカード33等の外部記憶装置などを制御するI/Oコントロール回路部25、前記ノートパソコンの外部に別の機器等を接続するための拡張バス部26を有する。

【0055】図3のCPUモジュール2の外形図を図4に示す。又、図3のCPUモジュール2、メモリ・CPUコントロール部7、メインメモリ27、BIOS ROM24等が実装されたCPU基板46の外形図を図5に示す。本実施例のノートパソコンは、裏蓋を開けると図5に示した基板46を、ユーザーが触れられるようになっている。ユーザーは裏蓋を開け、自ら増設RAMモジュール47を2枚増設することと、CPUモジュール2を交換することができる。

【0056】図4に示すようにCPUモジュール2はCPU基板46から自由な脱着を可能とする為のコネクタ40とCPU1で構成される。CPUモジュール2上につくCPU1の外形は、PQFP(PLASTIC QUAD FLATPACK PACKAGE)、PGA、LGAのいずれでもよく、又それ以外のどのようなパッケージでもよい。パッケージのピン数も自由である。CPUの外形がPGAの場合をCPUモジュール41に、

17

LGAの場合をCPUモジュール42に示す。CPUモジュール2という形にすることで、安価で小型薄型のQFP、PQFPというパッケージを使用することができると共に、将来的な新パッケージや異なるメーカーのCPUに対しても対応することができる。ノートパソコンのように限られたスペースしかないものでは特に有効である。

【0057】本発明の代表的な概略のフローチャートを図6にブロック図を図1に示す。本発明は、従来のパワーセーブを発熱対策に応用し、CPUもしくはケース内温度と、CLK周波数及び個々の周波数における動作時間と、CPUモジュール2から通知されたCPUタイプと、ユーザーによる動作モード設定等により温度制御を行なうものである。そして、その方法は個々の温度設定もしくは他の設定値によってCPUのCLK周波数を切り換えたり、キャッシュ制御を切り換えたりCPUをHOLD状態にしたり、CPUの電圧を変化させたりすることによって行なう。他に、図11に示すCPUのような場合は、CPUコアに対するCLKを一時的に停止するSTPCLK端子を利用し、一定時間ごとにCPUコアに対するCLKを止めることで見かけ上CPUのCLK周波数を落とすと共にCPUの消費電力を落とす事もできる。又上記制御条件、方法はすべて行なわれなくともよく選択的に応用できる。つまり本発明は、例えばCPUモジュール2でなく図2の(B)に示すようなICソケット20を有するデスクトップPCにおいて温度上昇時にCLK周波数をおとす場合にも有効である。又、図6のフローチャートにおいても各々のステップをすべて必要としない。このように本発明は多くの場合がありそのすべてを説明するのは難しい。従って以下に図6のフローチャートを基に各々のステップごとに詳細に説明するが本発明は各項目のどの組合せによる制御方法においても有効であるに注意されたい。

【0058】1. ステップ50

ユーザーはパワースイッチをONにする。

【0059】2. ステップ51

本ステップでは、CPUのタイプを判別する。本ステップはあらかじめCPUが固定でありグレードアップされない場合必要ない。CPUの形状、タイプ及びその判別方法にはいくつか種類があるので以下に説明する。

【0060】(1). CPUの形状

先に述べたようにCPUの形状にはPQFP、PGA、LGA等がある。グレードアップ可能なPCを考えた時その実装方法として、図4のCPUモジュールの形にする場合と、図7のPGA形状のCPU60を基板上に実装されたICソケットに直接セットする場合、及び図8に示すように元々基板上に実装されているのはQFP等の形状のCPU62でグレードアップされる場合にはアップグレードソケット63にPGA形状等のCPUをセットする場合等がある。

18

【0061】(2). CPUの判別方法

CPUにどのようなタイプがあるかは次の項目で詳細に説明するとして、ここでは上記CPUの形状ごとにCPUの判別方法を説明する。

【0062】a). 第1に図4に示すCPUモジュール2の形態にする場合について図9を用いて説明する。図9はCPUモジュール2の簡略化した回路図である。CPUモジュール2は、基本的には、CPU1とコネクタ40で構成され、CPU1の入出力信号65は、コネクタ4と接続されている。ただし、CPU1とコネクタ4の信号数は同じ数ではなく、CPU1にある信号で必ずしも必要でない信号(例えばグランド線など)は省かれる。図1のSCK0信号76、SCK1信号75、SCK2信号74がCPU1のCLK周波数を示す信号であり、SINT信号72、S2XC信号73がCPUタイプを示す信号である。CPUモジュール2上では、ハンダジャンパー71によってCPU1の種類により各々の信号の値を'H'か'L'に選択することができるようになっている。又、CLK2信号67はCPU1の内部CLK周波数どうりのCLKを出力する信号であり、CLK68はCLK周波数の倍周波数のCLKを示す信号である。システムからは、常に両方のCLK信号が出力されているので、CPUタイプによってジャンパー66で切り換える。他に、CPU1がSMIに対応しているかどうかによって電源オフやポップアップメニュー表示、ソフトウェアパワーセーブ機能などによるインタラプト発生を図9のSMI69、INTR70のどちらかに選択的に供給することで、CPU1の機能によってそれぞれ最適なシステムを構築する。

【0063】このようにCPUモジュール2の形態にする場合、上記意外でもいくつもの設定をCPUモジュール2上のジャンパー71等で設定できるので応用範囲が幅広く1番優れた方法である。たとえば、インテル社の486SXからPENTIUMにグレードアップする場合、専用のICソケットを用いる事なく実装可能であり、CLK周波数ばかりでなくCPUタイプもジャンパーで設定することで本体に通知する事ができる。これはおよそ120万トランジスタを有する486と310万トランジスタを有するPENTIUMのどちらが実装されているかをハード的に本体に通知することで回路の切り換えを可能とするもので、放熱板ばかりでなく専用の空冷ファンまで必要とするPENTIUMをノートパソコンに利用可能とする本発明の重要ポイントの1つである(486SX、PENTIUMはインテル社の商標である)。

【0064】b). 第2に図7のPGA形状のCPU60を基板上に実装されたICソケットに直接セットする場合について説明する。この場合通常はあらかじめ決められたCLK周波数に一致するCPUのみによりグレードアップする。CPU側で内部のPLLによってCLK

周波数を何倍かにあげるか、CLK周波数は同じで内部回路を変更することで処理能力をあげる。温度制御のために本体がCPUのタイプを知る方法は、CPU内部レジスタによって判定するか、基板上に別に用意されたジャンパーをユーザーに変更してもらうことにより判定する方法などがある。

【0065】c). 第3に図8に示すように元々基板上に実装されているのはQFP等の形状のCPU62で、グレードアップされる場合にはアップグレードソケット63にPGA形状等のCPUをセットする場合について説明する。この場合もほとんどb)と同じである。通常図7のCPU60が図8のアップグレードソケット63にセットされた場合、CPU62の信号を解放するグレードアップ信号が用意されているので本信号を利用すればグレードアップされるCPUを限定した場合に温度制御、パワーセーブのためのCPUタイプ判別は容易である。

#### 【0066】(3). CPUのタイプ

CPUのタイプを判別することは本発明に於て非常に重要である。本発明ではグレードアップされるCPUによって温度制御、パワーセーブの手段を最適化することにより高いパフォーマンスを実現可能とする。温度上昇カーブはCPUによって異なるので、CPUのタイプを知ることがCPUをアップグレードする事を可能とする情報処理装置で温度制御を行なう場合、欠くことができない要素である。

【0067】以下順をおってCPUのタイプについて説明するが本発明では必ずしもすべてを判別しなくともよく選択的に判別する機能を持つことも可能である。

#### a). CLK周波数

最初にグレードアップされるCPUの情報として必要なのはCPUの最大CLK周波数である。これは同じCPU(たとえば486SX)でも周波数によっては温度制御を必要としない場合があるのでその切り分けに使用する。

#### 【0068】b). CPUの種類

次に必要なのはCPUの種類である。これは、実装されているCPUが486SXなのか、486DX2なのか、PENTIUMなのかを判別する為である。同じCLK周波数でもCPUが異なれば、温度上昇カーブが異なる。CLK周波数だけではトランジスタの数や構造が大幅に違うCPUにたいして一律の温度制御を行なうことは不可能である。

#### 【0069】c). インタラプトの種類

前述した、SMIがサポートされているか否かを判別する。SMIがサポートされている場合、ソフトウェアで温度制御、パワーセーブを行なうことができるためハードウェアで完全にコントロールするよりも幅の広い制御を行なうことができる。尚、SMIを使用しなくとも他のインタラプトを使用してソフトウェアで温度制御等を

行なうことができるが、プログラムによっては該当するインタラプトを使用している為、温度制御できなくなる場合がある。従って、通常はSMIがサポートされていない場合はハードウェアで完全にコントロールする。

#### 【0070】d). CLKタイプ

温度制御等では、主にCLK周波数を切り換える事によって行なう。従ってCPUのCLKタイプ、CLKの切り換え方の情報を得ることは非常に重要である。CPUのCLKタイプ別の内部ブロック図を図10、11、12に示す。

【0071】図10には標準的なCPUのタイプを示す。まず(A)に386SX等(386はインテル社の商標である)の内部ブロック図を示す。このタイプのCPUではCPUコアに直接CLK2信号67が入力される。このように外部CLKと内部CLKが同一であるためリアルタイムにCLKを切り換えることができる。

(B)は486SX、486DXなどのCPUの内部ブロック図であり、内部に入力周波数を2倍にするPLL81を有しておりCLK68の倍周波数のCLKがCPUコア80に入力される。(C)は(B)のPLL81が入力周波数を2倍にするのに対し、入力周波数をn倍にするPLL82を有している。486DX2ではこのPLL82が入力周波数を4倍にする。(B)、(C)に示すようなCPUではPLLで入力されたCLK周波数を数倍に上げるので、CLK周波数をリアルタイムに変化させることはできない。従ってこのようなCPUでCLKを切り換える場合、CPUにRESETをかけその間にCLKを切り換える等の方法しかない。この為、CLK切り換えに時間がかかるので頻りにCLKを切り換えるとパフォーマンスを著しく損なうことになる。このことは温度制御の方法を決定するのに重要な事である。又、(B)、(C)の外部CLK周波数が同じ場合(C)の方が内部CLK周波数が高い分だけ消費電力が増大する。従って、CLK68の周波数ばかりでなく、内部のPLL82が入力周波数を何倍にするかという情報も温度制御の為には重要である。これらタイプは前述したCPUモジュール2上のジャンパー71の設定等で本体に通知される。

【0072】図10の(B)、(C)では、前述したようにCLKを切り換える場合特殊な処理を行なう必要があるがこの機能をCPU側にあらかじめ持ったのが、インテル社のエンハンスドSL 486DX2であり図11であらわされる。実際は複雑なロジックであるがここでは簡略化して示してある。

【0073】PLL82から出力されるCLKを切り換える場合、まずSTPCLK83を'L'にしてCPUコア80に出力するCLKを止める。次にCLK68のCLK周波数を変更して、PLL82が安定するまで一定期間待つ。その後STPCLK83を'H'にしてCPUコア80に再びCLKを出力する。

【0074】ところが、この場合でもPLLを内蔵しているCPUでは、CLK切り替えの際にPLLの安定待ち時間分だけパフォーマンスを損なう。

【0075】そこで、このタイプのCPUではSTPCLK83を一定周期で所定時間CPUに入力することで見かけ上のCLK周波数を落とす方法も考えられている。そうすれば、外部CLK周波数を変化することなくCPUの消費電力を下げる事ができ、リアルタイムに見かけ上のCPUのCLK周波数を変化することができる。実際には、STPCLK83をある一定周期（周期は、レジスタなどで幾通りにも設定可能である）でCPUに対して入力する回路をもち、STPCLK83の幅（時間）もレジスタなどで幾通りにも設定可能にすることで実現できる。ただし本方法はCPUのコアを止めるタイミングを考慮しなければならず回路が複雑になる欠点がある。詳細は後述する実施例2で説明するが、CPUがSTPSLK83を受け実際に内部CLKを止めることを許可する信号を待たなければならない。そこで、現在のCPUでは図10、図11の構造しかないが、本特許は将来的なCPUに対しても有効であることを示すために、CLK切り替えの際にパフォーマンスを損なわない構造を持ちかつ外部CLK周波数を落とす為にPLLを内蔵したCPUの内部ブロック図の例を図12に示す。本例は一部を説明するだけで本発明は図12に示した以外の構造に対してももちろん有効である。

【0076】図12で3例を示す。（A）は外部からセレクト信号のSEL86を与えることにより、PLL82から出力されるCLKをCLKB85とセクタ84で切り換えてCPUコア80にCLKを供給する方法である。CLKB85が選択されたときは直接CPUコア80に入力されるのでCLK周波数をダイナミックに切り換えることができる。こうすればCPUコア80はCLK切り替えの際にも処理し続けるのでパフォーマンスを損なうことはない。また、CLK68の周波数を固定にしておき単純にCLK68とCLKB85を切り換えることも可能である。

【0077】（B）は（A）のCLKB85にPLL87を追加したものである。2つのPLLを有することで、CPUコア80のCLK周波数が高い場合で、切り換えるCLK周波数が両者とも高い場合でもCLK68、CLKB85の両方のCLK周波数を落とすことができる。（A）に比べてCLKB85側の周波数もおとすことができるのである。

【0078】（C）は（A）のCLKB85を持たず、PLL82の出力とCLK68の出力をセクタ84で切り換えるものである。図11に比べ、同じ端子数にも関わらず本方法に従えばCLK切り換え時にCPUコア80の動作を停止する必要がない。

【0079】以上のように、本発明では上記のさまざまなCPUタイプを判別しそのCPUに適した温度制御、

パワーセーブを行なうことができる。当然のことながら本発明は、CISCベースのインテル製CPUのみならず、RISCベースのCPUなど全てのCPUに対しても有効である。

## 【0080】2. ステップ52

本ステップでユーザーは温度制御、パワーセーブについての動作モードを設定する。尚、本ステップは電源投入時の一連の初期化作業の中で行なわれなくともよく、動作中にポップアップメニュー（本発明でいうポップアップメニューとは、キーボードのキーコンビネーションもしくは専用キーを叩くことによって、アプリケーションを表示している画面に上書きされる形で現われる各種設定画面のことである。）内で設定が行なわれてもよい。また本ステップを省略しシステム側で自動的に制御を行なうことも可能である。

【0081】各動作モードの詳細な説明はステップ53、54で説明する事として、ここでは概略の動作説明と、その動作モードの切り換えについてその必要性を説明する。図13に連続動作時のCPUをノートパソコンに使用した場合の表面温度上昇カーブを示す。温度上昇カーブ90、91、92、93の順でそれぞれ対応するCPUの消費電力が高くなって行く。従来のノートパソコン等で使用されているCPUでは温度上昇カーブ90、91のように連続動作をし続けても温度限界94（この場合80℃とする）には達しない。ところが放熱板と空冷ファン等の強制的な空冷手段を必要とするCPUをノートパソコンに使用した場合は、温度限界を越えてしまう。そして、温度上昇カーブ92、93を比較して分かるようにCPUの消費電力が大きくなればなるほど短時間で温度限界を越える。

【0082】そこで従来では図14に示すように温度制御ポイント95（この場合70℃とする）を設けて、設定値に達したところでCLK周波数をおとしそれでも温度上昇を続けて温度限界94に達したときにはシャットダウンすることでCPUを熱暴走から保護していた。この場合の温度上昇カーブを96に示す。

【0083】次に図15に本発明の温度制御、パワーセーブを取り込んだ温度上昇カーブ例を示す。本発明の温度制御、パワーセーブについてはいろいろなモードが存在するので詳細はステップ53、54で説明するが、温度上昇カーブとしては概略図15に示すカーブを描く。本発明では、環境温度、システムの動作状態、CPUのタイプ、CLK周波数、そして本ステップ52で設定される動作モードによってリアルタイムにかつ細かく温度を制御するので図13、14と比べ温度上昇カーブは絶えず上下する。

【0084】ここで上記の温度上昇カーブを比較しながら動作モードについて説明する。ユーザーは以下に述べる動作モードを電源投入時や、ハード的なディップスイッチ、キーボードコンビネーションなどによって画面に

表示されるポップアップメニュー内等で設定を行なう。

#### 【0085】(1)．自動温度制御モード

まず本発明のポイントである自動温度制御モードがある。本モードについての詳細はステップ54で説明するので、ここでは概略を説明するにとどめる。ここでいう自動温度制御モードとは、従来の技術の項ですでに述べた筆者出願の特開平4-354009号公報、特願平3-164657号に示したパワーセーブの手段を温度制御に応用したものである。特開平4-354009号公報の技術を応用した温度制御方法をダイナミック温度制御、特願平3-164657号の技術を応用した温度制御方法を電圧可変型ダイナミック温度制御と名付けここで簡単に説明する。

#### 【0086】(a)．ダイナミック温度制御

ダイナミック温度制御とは従来例のようにCPUの温度が上がってから制御するのではなく、電源投入時からCPUの温度上昇を極力抑えることにより長時間CPUのパワーを引きだせるようにしたものであり、特開平4-354009号公報に示されているパワーセーブの方法（ここではダイナミックパワーセーブと呼ぶ）と密接な関係がある。

【0087】ダイナミックパワーセーブとは、ある特定の要因（特定のIOアクセス、特定のインタラプト）が生じた場合、CPUのCLKの周波数はあらかじめ決められた時間、高周波数になり、要因が無い場合低周波数になるパワーセーブの方法である。ダイナミック温度制御では温度制御が必要なCPUの場合まずダイナミックパワーセーブに入る。これにより、CPUが処理を行っているときのみCLK周波数を高速とするので次に述べるノーマルモードよりも温度上昇が抑えられる。

【0088】ダイナミック温度制御では、ステップ53で得られた温度とステップ51で得られたCPUタイプ等から、温度上昇率を抑えるようあらかじめ決められた一定期間内のCLK周波数の高周波数時間を制御する。詳細はステップ54で説明するが、ダイナミック温度制御では一部パフォーマンスを損なうことがあることに注意されたい。

【0089】(b)．電圧可変型ダイナミック温度制御  
電圧可変型ダイナミック温度制御とは、上記ダイナミック温度制御がCLK周波数を制御するのに加えて電圧も変化させる温度制御方法である。パワーセーブにおいてダイナミックパワーセーブに対して電圧も変化させるパワーセーブ法は特願平3-164657号にある。このパワーセーブ手段を電圧可変型ダイナミックパワーセーブと呼ぶ。

【0090】電圧可変型ダイナミック温度制御では温度制御が必要なCPUの場合まず電圧可変型ダイナミックパワーセーブに入る。そして、ステップ53で得られた温度とステップ51で得られたCPUタイプ等から、温度上昇率を抑えるようあらかじめ決められた一定期間内

のCLK周波数の高周波数時間及び電圧を制御する。詳細はステップ54で説明するが、ダイナミック温度制御と同じく一部パフォーマンスを損なうことがあることに注意されたい。

#### 【0091】(2)．ノーマルモード

ノーマルモードとは、パワーセーブ、温度制御を行わずCLK周波数を該当するCPUの最大CLK周波数に固定する通常モードの事をいう。本実施例の自動温度制御は一部分必ずパフォーマンスを損なう。STEP54で後に詳しく説明するが、たとえば、長時間の計算を行うプログラムを走らせた場合前述の自動温度制御モードは必ずCLK周波数が低周波数になる。なぜならば、単純ループの場合でも高速になるような要因（たとえばメインメモリアクセスやVIDEO RAMアクセス）を、CLKを高速にする要因に入れてしまうとWINDOWSばかりでなくMS-DOS上のプログラムでさえ常に高周波数のCLKで動作してしまう。結果として、温度限界が短時間で訪れ温度制限のためにCLK周波数を落とさざるを得ない。それを回避するため本発明の自動温度制御は、キーボード、マウスなど外部から物理的な信号が加わった場合のみを要因として温度制御を行なうことで温度限界が短時間で訪れるのを防ぐ。従って長時間の計算を行うプログラムを走らせた場合、自動温度制御モードによってCLK周波数が落とされてしまう。

【0092】このことは一部のユーザーにとって非常に困ることである。自動温度制御モードは一般的なアプリケーションプログラムではパフォーマンスを損なわないことを前提に設計しているので、一般的なアプリケーションプログラムでは自動温度制御モードによってCLK周波数が落とされることはまずないといえるが、科学計算など自作のプログラムを作成し実行しようとしたときに自動温度制御モードによってCLK周波数が落とされてしまうのはCPUパワーを有効に使用できるとはいえず、せっかくのCPUパワーがむだになってしまう。そこで、キーボードのキーコンビネーションもしくは、キーコンビネーションか専用キーの殴打によって現われるポップアップメニューによって、ノーマルモードにする手段を用意する。キーコンビネーションとは、たとえばCTRLキーとGRAPHキーとPキーを同時に押すことを意味する。ノーマルモードでは、ステップ53で計測されるCPUの温度が限界に達したら、CLK周波数をおとすなどしてCPUの温度上昇を自動的に防ぐ。一度温度が上昇すると冷えるのに時間がかかるので、その後のパフォーマンスは自動温度制御モードの時に比べて落ちる。

【0093】このように、両モードともメリット、デメリットがあるので必ずユーザーに選択する手段を設けることが必要である。これは本発明の重要なポイントの1つである。他にたとえば10分～1時間をこえるような

10

20

30

40

50

計算の場合上記両モードで動作させるよりも常に温度限界点で動作させるのがよい場合もある。この場合、あらかじめ計算されたCLK周波数以下にCLK周波数を落とすことによってそれを実現する。すなわち、本発明の情報処理装置では、CLKの周波数を落とした場合、それが連続動作可能な周波数の場合は自動的に温度制御をはずす機構も備えている。これは、BIOSがCLK周波数を設定する時に温度制御をはずすことによって行う。前記STPCLK83を有するCPUの場合でCLK周波数を変化させずSTPCLK83を制御する時は、STPCLK83の周期およびパルス幅がある一定条件（CLK周波数の平均値が連続動作可能な周波数と同じ時に相当する）にあった時、自動的に温度制御をはずす。尚、本モードは上記2モードで代替可能なので必ず必要であるわけではない。CLKの周波数を落とした場合、自動的に温度制御をはずす機構を備えることも本発明の重要なポイントの1つである。ここで以上のことを分かりやすく下記に示す。

【0094】(a)．通常アプリケーションの場合  
自動温度制御モードを使用

(b)．1秒程度～10分程度の処理の場合  
ノーマルモードを使用

ユーザーは処理後、自動温度制御モードにもどす。

(c)．10分程度以上の処理の場合  
CLK周波数を落とし温度制御を解除して使用する。

【0095】尚、後述する実施例3では、上記(a)～(c)までを自動的に切り換える手段を説明する。実施例1では、ユーザーがメニュー等でモードを切り換えることを前提に説明する。

【0096】3．ステップ53

本ステップでは温度を計測する。本発明の温度計測方法には、CPUの表面温度を直接計測する方法と、ケース内部の温度を計測する方法及び、ケース外部の温度を計測する方法がある。他に、センサーを設けず電源投入時からのCLK周波数の割合から温度カーブを推定することも可能である。この場合はCLK13を生成、制御を行うメモリ・CPUコントロール部7内にカウンターを設け、あらかじめ決められた一定時間でCLK周波数もしくはCLK停止信号を監視しカウントすることで行う事が出来る。たとえば、自動温度制御時にはCLK周波数保持時間が1mS～数S単位で変化するので、100μSごとに周波数を監視して、最大周波数時はカウントアップし最低周波数時はカウントダウンするようにする。そしてカウント値が1000000に達すると（高周波が100S連続することに相等する）温度が上昇したと判断できる。

【0097】この手段が、請求の範囲で示した周波数別動作時間に相当する。概要は従来の技術の項でも述べたが、本実施例ではCPUのタイプ、動作周波数がわかるようになっているので、各々のCPUに対して何%の動

作により温度限界に達するかをすることができる。（その%を動作限界%とする）常にスピードが動作限界%に達しないよう制御すれば温度限界が訪れない事になる。自動温度制御時などCPUは常時高周波数で動作しているわけではないので、上記カウンタが動作限界%になることはまれである。CLKを高速にする要因が連続に発生したときには動作限界%を守るよう自動的に動作周波数を制御する。（ただし、従来技術の項で述べたように環境温度等の考慮が不十分である）又、実際の計測した温度と上記カウンタ利用による、動作限界%を守るよう自動的に動作周波数を制御する手段を組み合わせ、ある温度に達するとカウンタ制御である動作%におさえることもできる。

【0098】尚、本ステップ53は実際には図6に示すようなシリアルな処理ではなくリアルタイムに処理を行っていることに注意されたい。以下それぞれの温度計測方法について説明を行なう。

【0099】(1)．CPUの温度を直接計測する

本発明は、CPUの温度を制御するのだから、CPUの温度を直接計測する方法がいちばん適している。この場合2つの方法が考えられる。1つ目はCPU自身が内部に温度センサーを持ち自らが保護するか、警告信号としてあらかじめ決められた信号を出力する方法がある。もう1つはCPUの表面にセンサーを付ける方法である。

【0100】(a)．CPUの内部にセンサーがある場合

CPUの内部にセンサーを設けて自らの温度限界時にCLK周波数を強制的に下げるなどして保護する方法が考えられる。この場合当然のことながらこのような構造を有するCPUのみしか対応できない。CPUは自らの温度限界を知ることはできるが、現在の処理がプログラムがどのようなことを意図して処理を行なっているかをすることはできない。したがってCPUの内部にセンサーを設ける場合、熱暴走のための保護をかけることはできるが動作途中で細かい温度制御を行なうことは難しい。尚、外部からの温度制御信号を取り込み制御するのであればセンサーがどこにあるかの違いだけで外部にあるのとかわらないので細かい温度制御を行なうこともできる。CPU内部にセンサーがあると(b)で述べる方法よりも高さを抑えられると考えられるので実装上是適している。

【0101】(b)．CPUの表面にセンサーを付ける方法

本実施例を図2に示した。(A)はQFP等のフラットパッケージのCPU1の上にサーミスタ3を付けた方法である。(B)はPGAのICソケット20のまん中にサーミスタ3をあらかじめセットし、図7のようなPGAのCPU60をセットするとCPUにサーミスタ3が接触する（もしくは近傍に配置される）ようになっている。

【0102】ノートパソコンのようにケース内部の高さ制限が厳しい場合(B)の形よりも(A)のようなフラットパッケージのCPUが使用される。そしてこのCPUがグレードアップされる場合、図4に示したCPUモジュール2の形になる。故にサーミスタ3がCPUモジュール2上のCPU1の上にセットされる事になるので(B)の場合よりは高さは低くなるものの、サーミスタ3の分だけ高さが高くなるという欠点がある。また、ユーザーが交換するCPUモジュール2上にセンサーがく

ことで物理的に不安定であるという問題もある。  
【0103】この問題を解決する一手段を図5に示した。CPUモジュール2上のサーミスタ3はCPU1の端子からでたグランドパターン48(電源パターンでも可)上にセットされる。CPU1の発熱が端子からグランドパターン48に伝わりサーミスタ3で感知される。サーミスタ3は熱電対など他の温度センサーでもよい。本方法は本発明のポイントの1つである。

【0104】さて、上記問題点を解決するもう1つの方法として、CPUの表面温度ではなくケース内部温度を計測する方法を次に示す。

【0105】(2)、ケース内部温度を計測するCPUの表面温度を計測するのが一番よい方法であるが、高さ及びユーザの取扱い上の理由で物理的な欠点があることを説明した。そこでここではケース内部温度を計測することでCPUの表面温度を計測するのと同等の結果を得ることができるかどうかを説明する。本発明に於て、本項目も重要なポイントである。

【0106】まず図16と図17に環境温度の違いによる温度上昇カーブを示す。図16は環境温度が25℃の場合のグラフであり、図17は図16とまったく同一のCPUで同一の動作条件下に於て環境温度が40℃の場合のグラフである。各図には環境温度そなわ室温102と、ケース内温度101及びCPU表面温度100の温度グラフを示してある。図16ではほぼ温度が飽和した4時間後の温度は、ケース内温度が室温より15℃高い40℃であり、CPU表面温度が室温より30℃高い55℃である。そして室温が25℃より15℃高い40℃の場合では図17に示すように、ケース内温度が室温が25℃の時より15℃高い55℃であり、CPU表面温度も室温が25℃の時より15℃高い70℃である。このように、ケース内温度及びCPU表面温度は、環境温度の変化分だけ温度上昇する事が分かる。特にケース内温度は、環境温度の上昇分を反映することに注意されたい。

【0107】次に、図18、図19にCPUの違いによる温度上昇カーブを示す。両グラフは室温25℃で同じ動作状態で計測したもので図18のCPUよりも図19のCPUの方が処理能力が高く従って消費電力も多い。当然の事ながら図19のケース内温度101及びCPU表面温度100共に図18よりも高い。ここで注目する

のがケース内温度101とCPU表面温度100の関係であり、両グラフからケース内温度がCPUの表面温度上昇を反映しているのが分かる。

【0108】以上述べたように、ケース内温度は、環境温度とCPUの表面温度の両方を反映する事が分かる。更に言えば、ケース内温度にCPUタイプ、動作条件を加味したある一定の温度を加算すればCPUの表面温度を推定することができる。本発明の実現手段の一つは、ケース内温度にステップ51で得られるCPUタイプに対応するあらかじめ決められた一定の値を加算することでCPUの表面温度を推定し温度制御を行なうことである。

【0109】実際にセンサーをセットする例を図5に示した。このようにサーミスタ等の温度センサーをCPU基板46上に備えればよい。こうすればCPUモジュール2が交換されても常に同じポイントのケース内温度が計測できCPU1の表面温度を推定できる。

【0110】(3)、ケース外部の温度を計測する上記(1)、(2)の方法の他にケース外部の温度を計測する方法もある。しかし、上記(1)、(2)の方がCPUの温度をより正確に制御可能であるので本方法は可能であると言うだけにとどめる。

【0111】4. ステップ54

本ステップでは、STEP51～STEP53で得た各情報に沿って温度制御を行なう。STEP52でユーザーが設定するモードのうち自動温度制御についてここで詳細に説明する。尚STEP52で説明した従来動作と同様のノーマルモードに対してはここでは説明しない。

【0112】STEP52で述べたように自動温度制御は従来のパワーセーブ、特に筆者出願の特開平4-354009号公報、特願平3-164657号に示したパワーセーブと密接に関係する。そして、自動温度制御には上記公報に対応してダイナミック温度制御と電圧可変型ダイナミック温度制御の2種類がある。そこで従来のパワーセーブについても若干の説明を加えながら自動温度制御について説明する。

【0113】自動温度制御についての概略はSTEP52で述べたのでパワーセーブとの関連性と、違いを示しながら実施例を説明する。図1に於て、メモリ・CPUコントロール部7が自動温度制御機能を有している。自動温度制御機能は大きく分けて4つのブロックに分けることができる。即ち動作モード設定部8、パラメータ設定部9、要因検出部10、CLK制御部11である。以下それぞれの機能に付いて説明する。

【0114】(1)、動作モード設定部8

動作モード設定部8ではSTEP52でユーザーが動作モードを設定した場合その設定によってパラメータ設定部9、CLK制御部11に対してコントロール信号を出力する。また、ダイナミック温度制御と、電圧可変型ダイナミック温度制御の設定も決められたI/Oポートア



クセスによって本ブロックで行なう。I/Oポートアクセスは図20のAに示すように、例えば101hのbit0に電圧可変型ダイナミック温度制御をする場合は1を、また電圧可変型ダイナミック温度制御をしない場合は0を書くことで行なう。

#### 【0115】(2)．要因検出部10

要因検出部10では、CLK周波数を高周波数にする為の要因を検出しCLK制御部11にコントロール信号を出力する。その要因はパワーセーブ時と自動温度制御時とは異なる。それは、前述の従来例の項で述べたような「WINDOWSのように組み込まれたデバイスドライバが一定間隔でデバイスをチェックするプログラムの場合、結果的に常に連続動作が発生することになり実際にCPUパワーが必要なときに温度限界によって低速動作になってしまう」事を避けるためである。この点は本発明のポイントの一つであるので以下更に詳細に説明する。ただし、これはCLK周波数保持時間の設定とOS（オペレーティングシステム）によって異なり要因をパワーセーブ時と自動温度制御時で同一にしても、WINDOWSにおいても問題ないこともある。ここではより

広い範囲で対応する実施例として説明する。

【0116】まず従来のパワーセーブの手法の1つであるダイナミックパワーセーブの動作波形を図22に、又電圧可変型ダイナミックパワーセーブの動作波形を図23に示す。図22の(A)で、キーボード、マウス、FDD等の動作時にI/Oポートにアクセスした場合に出力されるコマンド信号120が検出されると、図1のCPU1に対するCLK13の周波数を切り換える周波数切換信号122が'L'から'H'になる。周波数切換信号122が'H'になるとCLK信号13は決められた一定期間(図22では20mS)CPU1の最大CLK周波数(図22では20MHz)で動作し、周波数切換信号122が'L'になるとCPU1の最低CLK周波数(図22では2MHz)で動作する。尚、周波数切換信号122が'H'の時と'L'の時のCLK13の周波数を設定できるものもある。ユーザーが連続でキーボードを操作した時はコマンド信号120が(B)に示すように連続で発生する。その場合でも必ず最後のコマンド信号120から一定期間(図22では20mS)経過後CLK信号13は低周波数に切り換わる。

【0117】図22の波形において周波数切換信号122が一定期間'L'であると電圧を落とすのが図23に示した電圧可変型ダイナミックパワーセーブである。周波数切換信号122が'L'になってから、一定期間経過後、電圧切換信号123が'L'になり結果として電源の出力電圧124が5Vから3Vになる。

【0118】図20にダイナミックパワーセーブ コントロールポートを示した。(B)の表110に示すように、本実施例ではCLKを高速にする要因として12種類

の要因を一つずつ割当てソフトウェアで有効無効を設定できるようにする。又、それぞれの要因が発生した場合どのくらいの期間高速動作をさせるか(前述のCLK周波数保持時間に相等する)をレジスタ120h、130hで設定するようになっている。当然のことながら要因の数、レジスタの構成、時間の設定等は本実施例に限定されないばかりでなく、本実施例でbit5に割り当てたメインメモリなどは、アドレスによって領域ごとに設定することも可能である。

【0119】ここで従来はあらかじめ決められたCLK周波数保持時間を電源立ち上げ時に設定していたが、本発明ではステップ51で得たCPUタイプ、CPUの最大CLK周波数によりこのCLK周波数保持時間を変化させることも可能である。ソフトウェアの動作を考えたとき、CPUのパフォーマンスが上げれば上がるほどその実行時間が短くなる。そこでCPUのパフォーマンスを知るために、ステップ51でCPUのタイプと最大CLK周波数を判別し、その結果をパワーセーブに反映させることでより適切な効果を得ることができる。本アイデアは本発明のポイントの一つである。

【0120】さて、従来前述した図22の波形で、低速期間121と高速期間125は要因にかかわらずその期間ごとの周波数を自由に設定できるものが知られていたが、本発明では図20の表111に示すように各要因ごとに最大周波数と最低周波数を設定できるようになっている。

【0121】図21に自動温度制御ポートを示した。表114を図20の表110と比べて分かるように要因が12から5つに減り設定時間が長くなっている。又、図20の表111に対応するのと同じように表115によって各要因ごとに最大周波数と最低周波数を設定できるようになっている。この点が本発明の重要なポイントの1つである。

【0122】尚、自動温度制御の場合もパワーセーブと同じくステップ51で得たCPUタイプ、CPUの最大CLK周波数によりこのCLK周波数保持時間を変化させることも可能である。

【0123】現在、表110に示したような要因が連続的に発生する場合、即ちCPUが連続で動作することで熱暴走を起こす可能性のある場合として大きく分けて以下の3つが考えられる。

【0124】(a)．デモンストレーションプログラム、WINDOWSなどのように一定期間ごとに各要因をアクセスした場合

この場合CPUは高速で動作し続ける。従って自動温度制御ポート114が表110のようになっていると温度制御ができなくなる。(実際の動作では、後述する実施例2に示すようにアプリケーション動作中は必ずしもCPU内部の処理をし続けるわけではないがここではあら

10

20

30

40

50

ゆるプログラムに対応できるようモデル化して説明をする。)そこでキーボード、マウスなど外部から物理的な信号が加わった場合のみを要因として温度制御を行なう。更に言うと、ユーザーがアプリケーションソフトに対し指示を与えた場合のみを要因として温度制御を行なうのである。

【0125】これを図24を使用して更に詳しく説明する。図1のCPU1のCLK13を切り換える周波数切り換え信号122は、表110に基づいて発生するダイナミックパワーセーブからの周波数切り換え信号130と表114に基づいて発生する自動温度制御からの周波数切り換え信号131のAND信号となっている。図24でWINDOWS動作中の期間136の場合信号130は'H'を保っている。この場合あらかじめ表114で設定される一定期間133(図24では5s)経過すると自動温度制御からの周波数切り換え信号131は'L'になり結果としてCLK13の周波数切り換え信号122は'L'となりCLK13の周波数は2MHzとなる。この後、表114で示される要因(たとえばキーボード入力)があった場合には自動温度制御からの周波数切り換え信号131はただちに'H'になり結果としてCLK13の周波数切り換え信号122は'H'となりCLK13の周波数は50MHzとなる。

【0126】以上のように実際にユーザーがキーボード入力などでアプリケーションの動作を促したときのみCLK13の周波数をあげるという制御を自動温度制御とすることによりWINDOWSなどのように一定期間ごとに各要因をアクセスするものでも温度制御を行なうことができる。尚、WINDOWSには、アプリケーションソフトの動作状態を本体に通知できる機能をもっているものがあるのでその場合はWINDOWS動作中136の期間でもBIOSからの周波数切り換え信号137によって周波数切り換え信号122が'L'になりCLK周波数が2MHzになる。このBIOSからの周波数切り換え信号137は図1の動作モード設定部8からCLK制御部11に対して出力される。

【0127】(b). 通信中、FDDのディスクコピー、HDDのフォーマット中、プリンタ動作中等の場合RS-232Cを介してモデムによって通信をしている時や、FDDのディスクコピー、HDDのフォーマット中等の場合、分単位をこえ場合によっては数時間におよぶ連続アクセスを行なう場合がある。この場合従来のパワーセーブでは常にCPU1が高速に動作していた。それ故、本発明で目指しているような連続動作では熱暴走してしまうようなCPUを使用する場合、ダイナミックパワーセーブ機能を有効にしても通信中に熱暴走を起こしてしまう。これを回避する為に(1)で説明したような「キーボード、マウスなど外部から物理的な信号が加わった場合のみを要因として温度制御を行なう」方法を用いると、パソコン通信などでダウンロード中な

どキーボード等を長時間操作しない場合突然CPUが遅くなってしまうことが考えられる。

【0128】そこで本問題を解決するため、各要因ごとに図21の表115に示すように最大周波数と最低周波数を設定できるようにする。尚、通常は最低周波数は固定してもよいので要因ごとに最大周波数のみを設定してもよい。表115においてFDD、RS-232Cだけは最大周波数を33MHzとし、その他の要因は最大周波数を50MHzとする。この場合の33MHzという周波数はCPUが連続動作しても熱暴走しない最大CLK周波数に相当する。図24において通信中135の期間はダイナミックパワーセーブからの周波数切り換え信号135は連続的に'H'になる。又、自動温度制御からの周波数切り換え信号131も連続的に'H'になり結果としてCLK13の周波数切り換え信号122は'H'となりCLK13の周波数は33MHzで動作し続ける。通信中にキーボード入力があった場合は図24では5s間は50MHzにCLK周波数があがる。

【0129】本方式は通信中のパフォーマンスがCPUのパフォーマンスよりも格段に遅いモデムのボーレートで左右されることと、FDDアクセス時のパフォーマンスがCPUのパフォーマンスよりも格段に遅いDMAコントローラの転送レート、FDDのアクセス時間に左右されることを利用して考えられている。すなわち、あるデバイスの連続アクセスが発生した場合そのパフォーマンスに対しCPUのパフォーマンスが寄与する割合が飽和することを考慮に入れたものである。

【0130】例えば、モデムから通信データを電話回線を通じて読み込み、画面に表示するまでを考えてみる。ここでCPUが関与するのは、ソフトウェアが行なう動作すなわちモデムからのデータをRS-232Cを通じて読み込みVIDEO RAMに書き込むまでである。モデムの通信が2400bps等の単位のシリアル転送で行なわれるのに比べ、CPU側は32bit、64bit等のパラレル転送を数十MHzで行なうことができる。つまり、このような一連の動作を考えた場合、CPUのパフォーマンスをいくらあげても、モデムの通信速度が全体のパフォーマンスのボトルネックとなってしまう実際のユーザーに対する体感速度に寄与できない。それならば、CPUのCLK周波数を、あらかじめ計算されたユーザーに対する体感速度に影響を及ぼさない程度のCLK周波数に落としてしまおうというのが、本アイデアである。

【0131】図20の表111においてダイナミックパワーセーブでも本自動温度制御方式と同じく各要因ごとに最大CLK周波数と最低CLK周波数を設定できるようになっていたが、この場合の最大CLK周波数はCPUのパフォーマンスが寄与する割合が飽和する点であるといえる。(前述のように最大CLK周波数のみを設定するようにしてもよい)これと同じく自動温度制御に

おいても図 24 の通信中 135 の CLK 周波数を CPU の温度限界ではなく、CPU のパフォーマンスが寄与する割合が飽和する CLK 周波数としてもよい。本実施例は RS-232C、FDD を例に記述しているが、ここで述べた方式を、RS-232C、FDD ばかりでなく CD-ROM や HDD、セントロなどの I/F を経由する機器（プリンタ他）等の他のデバイス一つ一つに応用することで全体的なパフォーマンスを落とすことなくパワーセーブ、温度制御を行なうこともできる。このことは本発明に於て重要なポイントのひとつである。

【0132】(c)．WINDOWS 中マウスを動作し続けた時などの場合

さて上記 2 種類の場合以外にどうしても CPU の温度限界に達してしまう場合がある。それは、WINDOWS 中マウスを動作し続けた時や、キーボード入力を連続で行なうことで CPU が最高速で動作し続けた場合である。又、STEP 52 でユーザーが動作モードをノーマルモードに設定し、CPU に対して最速動作を要求した場合も同様である。いままでは、従来技術の項で述べたように、温度センサーかもしくは温度センサーを持たず、ある一定時間の CLK 周波数の高周波数と低周波数の割合から温度上昇カーブを推定して現在の温度を知ることによって CLK 周波数を制御していた。本発明では STEP 53 で得られた温度を基に CLK 周波数をきめ細かく制御する。この場合については次のパラメータ設定部 9 で詳細に説明する。

【0133】図 1 において、キーボードコントロール部 22 から出力される信号 15（もしくは CPU がキーボードコントロール部 22 をアクセスしたことを示す信号の場合もある）や、マウス、FDD 等表 110、114 に示した各要因をコントロールする I/O コントロール部 25 から出力される信号（もしくは CPU が I/O コントロール部 25 をアクセスしたことを示す信号の場合もある）等に基づいて、要因検出部 10 では上記の各場合に最適なコントロール信号（周波数切り換え信号 130、131 など）を CLK 制御部 11 に出力することで CPU の CLK 13 を制御する。

【0134】(3)．パラメータ設定部 9

パラメータ設定部 9 ではステップ 53 で行なわれる温度チェックで得た温度と、各条件によるパラメータの設定値及び要因検出部 10 からの要因によって CLK 制御部 11 に対してコントロール信号を出力する。温度チェックの方法は、ステップ 53 で述べたようにセンサーを用いないで、あらかじめ決められた一定時間（たとえば 5 分）のあいだで、CLK 周波数を監視しその平均的な CLK 周波数から現在の温度上昇値を推定し、5 分後の CLK 周波数を決定するなどの方法もある（本発明はこの場合にも有効である）。

【0135】しかし、ここでは環境温度等に対応可能なセンサーを用いる方法である図 1 に沿って説明する。温

度チェックは図 1 の A/D コンバータ 5 からの出力信号 12 によって判断される。サーミスタ 3 は温度によって抵抗値が変化する。そこで、電源電圧 Vcc6 が、温度によって変化するサーミスタ 3 の抵抗値と抵抗 4 で分圧され A/D コンバータ 5 に入力される。A/D コンバータ 5 で電圧がデジタル値（本実施例では 3 ビット）に変換され、出力信号 12 としてメモリ・CPU コントロール部 7 内のパラメータ設定部 9 に出力される。

【0136】パラメータ設定部 9 で CLK 制御部 11 をコントロールするのは、前述の (2) の (c) に示したように CPU を連続的に最大 CLK 周波数で動作させた場合で CPU が温度限界に近付いた場合である。従来は、CPU の温度限界で CLK 周波数を単純に落とすことによって CPU の表面温度を下げていたが、本発明では更に 1 ステップみ込んで温度制御を行なう。

【0137】本実施例のパラメータ設定部 9 では、数段階の温度を設定しその温度によって最大 CLK 周波数を落とす。従って、温度によって要因検出部 10 で設定された最大 CLK 周波数に対し制限をかけることになる。

【0138】図 25 に温度によって最大 CLK 周波数を設定する表を示した。この表で CPU 温度によって CPU の最大 CLK 周波数が決定される。実際には、STEP 51 で得られる CPU タイプ（図 1 では TYPE 14 に相当する）によって、STEP 53 で得られる各温度設定（CPU 温度の場合やケース内温度の場合もある）に対応する CPU の最大 CLK 周波数を BIOS で設定する。図 25 で CPU 温度が 70℃ の場合、要因検出部 10 で図 21 の表 115 に基づいて最大 CLK 周波数が 50 MHz に設定されていてもパラメータ設定部 9 によって強制的に 40 MHz に CLK 周波数を落とす。

本実施例の CPU の温度規格を守る最大連続動作 CLK 周波数は 33 MHz なので、CPU の温度限界 80℃ に近付いたとき（ここでは 75℃）には 33 MHz に CLK 周波数を落とし連続動作でも CPU の温度規格を守るようにする。又、ユーザーが環境を機器保証外で使用することも考慮し、温度限界 80℃ の場合では更に CLK 周波数を落とす機能も有する（図 25 では 25 MHz にしている）。

【0139】実際の温度上昇カーブを図 26 に示した。図 26 で、ポイント 140 において CPU の温度が 65℃ に達する。そうすると要因検出部 10 及び動作モード設定部 8 の両方からの各周波数切り換え信号 130、131、137 の AND 信号である周波数切り換え信号 122 が 'H' の途中でも CLK 周波数 132 は 50 MHz から 45 MHz に落ちる。キーボード入力がなくアイドル状態が続くと図 26 のように温度も徐々に低下するので、再び CPU が動作する時には 50 MHz で動作することができる本実施例では、温度によって最大 CLK 周波数を制御する例を示したが、他に温度によって表 114 の各要因による時間設定をかえる方法もある。尚、

さきに述べた電圧可変型ダイナミック温度制御の場合は、温度によってCLK周波数ばかりでなく電圧も下げることが可能である。

【0140】又、ケースに強制空冷手段として空冷ファンを設けた場合、本ステップにおいて空冷空冷ファンの回転数を制御する事ができる。従来でも、空冷ファンの回転数を環境温度で変化させる例はあったが、本発明では、自動温度制御に空冷ファンの制御を加えることでより緻密な温度制御を行うことができる。

【0141】その一実施例では以下の順で制御される。  
①. 温度上昇し温度限界に近づくまで自動温度制御を行う。この場合最大CLK周波数は変化させないので通常使用でのパフォーマンスの低下はない。空冷ファンは必要ないので停止する。

【0142】②. 温度限界に近付いたとき  
自動温度制御はそのまま継続。最大CLK周波数は変化させない。空冷ファンを回転し温度によって回転数を制御する。

【0143】③. 更に温度限界に近付いたとき  
自動温度制御はそのまま継続。最大CLK周波数を落とす。空冷ファンを回転し温度によって回転数を制御する。

【0144】本例では従来のように温度により一律に空冷ファンの回転数を落とすのに比べ、オフィスに許される騒音と、ケースの大きさによって決定される空冷ファンに合わせて詳細にシステムを構築できる。その結果、ワープロなどのようにCPUパワーを殆ど使用しないアプリケーションソフトウェアの場合空冷ファンの騒音に悩まされる事なく操作でき、又ノートパソコンに装着されるような非力な空冷ファンであっても限界温度を越えた場合の温度制御手段を別に持つことにより、その空冷ファンの効果を最大限に利用する事ができる。空冷ファンがあれば前述のノーマルモード時の動作時間をより延ばすことができるのはもちろんである。

【0145】(4). CLK制御部11  
CLK制御部11ではいうまでもなく、動作モード設定部8、パラメータ設定部9、要因検出部10からのコントロール信号に基づいてCLK13の周波数を制御するブロックである。

【0146】5. ステップ55

ユーザーはパワースイッチをOFFにする。

【0147】以上が、本発明の温度制御方法であり、CPUのタイプ、ユーザーによる動作モードの設定に基づき、温度とソフトウェアの動作条件、種類によってCPUの温度を細かく制御する。

【0148】尚、いままでの説明は分かりやすくするためにある要因からCLK周波数を制御する方法に限定したが、他にインタラプト中に限り高速動作する手段等他の手段を組み合わせることも可能である。又、アドレスバスやメモリアドレスを監視することでプログラムのル

ープ性を認識しアイドル状態を検出する手段も本発明に取り入れることもできる。この場合、図20表110の内メインメモリの要因に本条件を加えると、計算中と単純連続ループを判別できより詳細に温度制御可能となる。

【0149】本発明は、従来の方法と異なり、あらゆる点でパフォーマンスを落とす事なく温度を制御する工夫がされており、かつCPUが何段階にもアップグレードされる場合にも有効である。本発明によって、従来不可能であると思われた放熱板と空冷ファン等の強制的な空冷手段を必要とするCPUをノートパソコンに使用することが可能である。現在、CPUはテクノロジーの進歩によってチップサイズの縮小化、電源電圧の低下が行なわれている。ソフトウェアはCPUの処理スピードが速くなればなるほど、実際にCPUが処理している時間は少なくなっていく。CPUがこの先、進歩すればするほど処理スピードが速くなるので、よりいっそう従来のソフトウェアがCPUを実際に使用する割合は少なくなる。本発明はこのことに注目して発明されているため将来に渡っての継続性を有する。

【0150】本発明は、ノートパソコンを例として述べたが、より小型のサブノートタイプ、パームトップタイプ、ハンドグリップタイプ、の情報機器においても応用可能であるばかりでなく、環境問題で問題になるデスクトップ型のコンピュータの消費電力低減にも応用可能である。デスクトップ型に応用する場合は、本発明に空冷ファンの回転制御を加えることでより実現的な制御を行なうことができる。

【0151】又、本実施例ではおもにCPUの消費電力をCLK周波数を制御する事で行なう例を述べてきたが本発明はそれに限定されるものではない。例えば、CPUの動作ブロックをソフトウェアの必要とするブロックだけ動作するように制御できるのが一番よい。数値演算プロセッサ部などその機能を使用するアプリケーションが必要なときだけそのブロックにCLK信号を供給すればよい。

【0152】将来的には、本発明をベースにソフトウェアとCPUの各ブロック動作が密接に連携できるよう、ソフトウェア、CPUを改良することによってよりパフォーマンスの高いCPUをより小型の情報機器に使用することができる。

【0153】又、前述のように、内部にPLLをもつCPUでCPUのCLK周波数を変化させるとPLLの安定時間待たなければならぬような場合は、CPUコアのみのCLKを停止させる信号を持ち、その信号を一定周期で所定時間CPUに入力することで見かけ上のCLK周波数を落とす方法もある。本発明は、CPUタイプを判別する手段も有しているので、そのCPUにあった手段で温度制御を行うことができる。CPUコアのみのCLKを停止させる信号をリアルタイムにコントロール

することで見かけ上、前記詳細説明のCLK周波数を変化させる手段と同一の効果を与えることができる。このことは、設計上外部CLK周波数を今以上、上げるのが難しい現在の状況下に於て、CLK周波数をPLL等により通倍化し内部のみCLK周波数をあげたCPUに対しても本発明が有効であることを示しており本発明の重要なポイントの1つである。

【0154】<実施例2>実施例1に基づき、より具体化した例を実施例2として説明する。実施例2における情報処理装置は実施例1と同じく図3に示すブロック図と同様な構成をもつパーソナルコンピュータ（以下パソコン）である。本パソコンは図11のSTPCLK83に相当する信号でCPUコア80のCLKを定期的に停止することでCPUの消費電力を下げると共に発熱を防止する機能を有している。その機能は大きく2つに分類される。第1は前述したノーマルモードである。本パソコンは図28に示すような、各コンパレータ151の出力信号H0～H3によってCPUの表面温度を推定し各値によってCPUコア80のCLKを定期的に停止する。第2は自動温度制御である。自動温度制御は実施例1で説明した方法と同様の機能を有している。本パソコンでは、自動温度制御をポップアップメニューによりON/OFFすることが可能である。自動温度制御がONでもコンパレータ151の出力信号H0～H3によってCPUコア80のCLKを定期的に停止する機能は有効である。

【0155】はじめに本パソコンのCPU表面温度検出部について説明する。図27はCPUモジュール2の外形図でありCPU1がついていない裏面から見た図である。本CPUモジュール2は処理能力の異なるCPUをCPU1として実装可能でありユーザーはCPUモジュール2を交換することでパソコンの処理能力のグレードアップを図る事が可能である。図28にCPU温度検出部の回路図を示す。サーミスタ3はCPUの温度限界によって決定される抵抗R4（146）と共に図27に示すようCPUモジュール2の中央部に実装される。その近傍にはCPUの温度をより反映させやすくするため穴141も設けて有る。サーミスタ3は穴141の中すなわちCPU1の裏面に直接接続してもよい。ここではユーザーが取り付けるので壊れにくくする為と、部品の取付の安易さから基板にサーミスタ3を実装する例をあげた。CPUの裏面は的確に表面温度を反映する為サーミスタ3は裏面の中央部に実装している。CPUの表面温度はCPUモジュール2の裏面に伝わりサーミスタ3に感知される。図28において抵抗R0（145）と、CPUモジュール2に実装される温度検出部147すなわちサーミスタ3の抵抗値+R4（146）でVccが分圧されコンパレータ151の+側に加わる。サーミスタ3は温度で抵抗値が変化するため、必然的に分圧された電圧も温度によって変化する。

【0156】ここでR4（146）がなぜ必要かを説明する。CPUモジュール2に実装されるCPU1はいくつかの種類が考えられるため、各々の温度限界値が異なる事が考えられる。あらかじめ特性のわかっているCPUの場合、実施例1の方法でCPUの種類を知り温度制御用のパラメータを変更することができるが、わかっていないCPUの場合対応できなくなる。そこでR4（146）をCPUの温度限界に合わせて変更する事で電圧の微調整を行って同一パラメータで制御を行う。又、CPUのAC特性を満たすため限定条件として限界温度幅が規定される事もありこの場合にもR4（146）があれば柔軟に対応できる。R4（146）を設ける事は本発明のポイントの1つである。

【0157】このように分圧された温度が反映された電圧値は、各設定温度に対応された電圧値と比較されその大小で'H'か'L'が出力される。たとえばH3では抵抗R1（148）と抵抗R2（149）の分圧比がH3の設定温度をあらわす基準電圧としてコンパレータ151の一側に加わる。この基準電圧よりも、+側に加わる電圧が上回る場合にH3に'H'が出力される。他の温度設定値も同様であり、H2の場合はR1とR3（150）の分圧比が基準電圧となる。以上の方法により各温度H0～H3のどの温度にCPU1の表面温度が達したか知る事が出来る。尚、H0に関しては後に説明する実施例3で使用するだけで実施例2では使用しない。

【0158】図30にノーマルモード時の温度上昇カーブを示す。本パソコンは設定温度H1（160）に達するまでは100%で動作し、H2（161）に達するまでは90%で動作する。又、H3（162）に達するとパソコンがどのような動作環境にあってもかならず温度がさがりようなスピードである50%に制限される。実際には、各段階ごとに設定するスピードはCPU1によってなるべくパフォーマンスを損なわないよう決定される。

【0159】本実施例2ではH1～H3の3段階で動作スピードを制限しているがCPU1によってはH3（162）のみでスピード制限をかけることも可能である。H3（162）のみで制限をかける例として実際のアプリケーション動作中の温度上昇カーブを図31に示す。

図31に示すように実際の動作ではCPU1は常にフル稼働するわけではなく、FDDアクセス時でHOLDがCPU1に入った時や、メモリ、IOアクセス時には稼働率が落ち、従って温度も下がる。実際に温度限界94に達するのは、連続してCPU1を使用する計算中や内部キャッシュで動作し続けるときなどごく限られた場合のみである。経験的には、DOSプロンプト状態などアプリケーションプログラムが動作していないときに温度が上昇する傾向にあるので（例えばキー入力検出ルーチンをCPUが繰り返すこと等が要因と考えられる）、CPU1によってはH3（162）で制限をかけても実

際の動作中には影響を及ぼす事はない。図 31 で、CPU1 は H3 (162) に達するまでは 100% で動作し、H3 (162) に達するとスピードが 50% (この値は CPU1 による) に制限され温度が下がる。温度が下がり H2 (161) に達すると再び 100% で動作する。

【0160】本実施例 2 では、H3 (162) に達したときに LED を消す事でユーザーに動作スピードが落ちた事を知らせる機能を有している。これは H3 (162) のみで制限をかける場合でも同様である。ユーザーはスピード制限がかかった事を知ることで、自動温度制御に切り換えたり、アプリケーションプログラムのアイドル検出プログラムを組み込んだり、動作環境温度を調整したりすることができる。これは本発明のポイントの 1 つである。LED を消す回路例を図 29 に示す。LED155 は RED と GREEN の 2 色を別々に発光するタイプの LED である。通常動作では、LEDOFF158 は 'H' になっており、LEDR156 によって RED が、LEDG157 によって GREEN が制御される。RED と GREEN が両方点灯すると ORANGE になる。本パソコンは CPU1 のスピードを変更する機能を有しており、最高速時には LEDG157 だけが 'H' となり LED155 を GREEN に点灯させ最低速時には LEDR156 だけが 'H' となり RED が点灯する。中間スピード時には、LEDG156、LEDR157 が共に 'H' となり ORANGE が点灯する。

【0161】LEDOFF158 は 2 つの AND ゲート 159 に接続される。図 30 で温度が上昇し H3 (162) に達する点 163 で LEDOFF158 は 'L' になり 2 つの AND ゲート 159 の出力が 'L' となるので LED155 は消灯する。温度が下がり H2 (161) に達する点 164 で再び LEDOFF158 が 'H' となり、もとに選択されたスピードを現す色に点灯する。H3 (162) と H2 (161) を数℃の間隔にすると、実際には温度が下降する時間は数 mS から数 S で下降するため、図 30 のように H1 と H2 を交互に行き来するのに従って LED が点滅するように見える。

【0162】次に、STPCLK83 を利用したスピード制御について具体的に説明する。実施例 1、3 でも共通の方法でスピード制御を行う事が出来る。図 33 に波形を示した。実施例 1 の STPSLK83 に相当する信号を STPCLKN165 として示す。STPCLKN165 は図 1 のメモリ・CPU コントロール部 7 から図 33 に示すようにパルス波形で定期的に CPU1 に出力される。その 1 サイクルを拡大したのが図 32 である。STPCLKN165 が CPU1 で認識されると CPU1 は内部 CLK169 を止める際に必要な処理を行った後、アドレスバスとステータス信号 167 に STPCLK ステータスを出力すると共に、アドレスストロブである ADSN166 を 'L' にする。そして、メモリ・

CPU コントロール部 7 は ADSN166 によって CPU1 が内部 CLK169 を止める準備ができたと判断し RDYN 信号 168 を 'L' とする。CPU1 は RDYN168 を CLK68 の立ち上がりでサンプルしそこから内部 CLK169 を止める。一定期間 STPCLKN165 を 'L' にした後再度 STPCLKN165 を 'H' にすると CPU1 は再び内部 CLK1 に CLK を供給すると共に必要な処理を行った後、通常動作に復帰する。上記の方法を用い、STPCLKN165 の 'H' の期間と 'L' の期間を可変にすることで CPU1 のスピードを自在に変える事が可能である。

【0163】通常、図 34 に示すように CLK68 は STPCLKN165 が 'L' の期間も周波数を変化させない。それは CLK68 を変化させると実施例 1 で述べたように PLL82 の安定期間分パフォーマンスを損なうからである。しかし PLL82 の安定期間は 1 mS 程度なので、ある程度パフォーマンスを損なっても支障がない場合、CLK68 を STPCLKN165 を 'L' にすると同時に周波数を落としてやれば CLK68 の周波数を変化させない場合に比べ消費電力をよりいっそう落とす事が出来る。その例を図 35、36 に示す。図 36 では、STPCLKN165 を定期的に 'L' に落とし、その 'L' の期間 CLK68 の周波数も落とす例を示している。又、図 36 を一部拡大したのが図 35 である。CLK68 の周波数を変化させた場合、STPCLK165 を立ち上げる場合は PLL の安定期間 Tp170 を経てから上げる点に注意する。PLL の安定期間 Tp170 は CPU1 によって時間を設定可能になっている。

【0164】本例では、CLK68 の周波数を落として消費電力を落とす方法を述べたが、更に消費電力を落とすためには、CPU1 の出力バッファをハイ・インピーダンスにする信号を STPCLKN165 が 'L' の期間入力すればよい。そうすれば、出力バッファが外部デバイスを駆動する電力もセーブすることができる。タイミングとしては、図 32 で RDYN168 を CLK68 の立ち上がりでたたく点から STPCLKN165 を 'H' にするまで、すなわち内部 CLK169 が停止している期間に CPU1 に入力する。CPU1 の出力バッファをハイ・インピーダンスにする信号は CPU1 のグレードアップを目的として本来 CPU に備わっていることが多い (高性能の CPU を追加した時、元の CPU の出力をハイ・インピーダンスにして、元の CPU を殺す為)。無論 CPU 外部に出力をプルアップする回路を設けても良い。

【0165】以上が実施例 2 の説明である。自動温度制御に切り替わっても、上記の温度検出の動作は同じなので要因が発生したときの最大周波数は温度によって制限される。このように統一的な制御にする事で制御回路を簡素化できる。

【0166】<実施例3>実施例1、2ではノーマルモードと自動温度制御モードをユーザーがポップアップメニューなどで切り換える例を示した。ユーザーがアプリケーションによって両モードを切り換えるにはある程度のプログラムの知識が必要である。そのため一般ユーザーにはどちらのモードで使用していいのかわからない場合がある。そこで実施例3で両モードを自動的に切り換える例を示す。

【0167】図37に温度上昇カーブを示す。本実施例ではH1(160)を使用せずH0(170)、H2(161)、H3(162)を使用する。まず、電源投入時はノーマルモードで立ち上がる。次にH3(162)に温度が達する点171で実施例2で示したように温度が下降してH2(161)に達する。そしてH3(162)まで再び上昇するというように上昇、下降を繰り返す。この期間(H3とH2の間に温度がある期間)がt1(175)を越える点172で動作モードが自動温度制御モードに切り換わる。長時間のCPU内部での計算中は要因が発生しないため動作スピードは最低速に固定され、温度は急激に低下しH0(170)を越えて最低速時のカーブ177に乗って下がっていく。ここでH0の設定ラインである170以下に温度が保たれる期間t2(176)が経過する時間174になると再び動作モードがノーマルモードへ切り換わる。このように、ノーマルモードと自動温度制御モードが切り換わる事で長時間の計算時もスピードをあまり落とさずに実行できる。

【0168】図37は長時間計算時の例を述べたが、他のケースでは以下ようになる。1つ目のケースはソフト的なアイドル検出プログラムが組み込まれていたときの場合である。この場合、立ち上げ時にはノーマルモードであるが、図31に示した実際の温度波形よりも更に低温側で温度カーブが推移するので実際にH3に到達する162のラインに達する事はまれである。t1を比較的長く設定すれば自動温度制御モードにはほとんど切り替わらない。次は、ソフト的なアイドル検出プログラムが組み込まれていない場合やあまり効果がない場合で、長時間の計算がほとんど行われない場合である。ユーザーがアプリケーションプログラム等で行う通常動作は、ほぼこのケースに該当する。この場合、ノーマルモードでは、一度H3のライン162に達すると、いざCPUのフルパワーが必要な場合に、100%のスピードで動作することが出来なくなる。それに対して本実施例3では、172のポイントで自動温度制御に切り換わるため通常の動作はほとんど172と173の間で行われる。又、実際プログラムが動作しているときは、ライン170を下回るt2の時間も短い(つまり多少の温度上昇を伴う)ので、ノーマルモードにはほとんど切り替わらない事になる。

【0169】以上のように両モードを自動的に切り換え

れば、あらゆる場合でもパフォーマンスをあまり損なう事無く動作することが出来る。t1をt2に対して大きくとれば、長時間計算時のパフォーマンスの低下はなくなっていくが、短期的なスピードダウンがt1の期間存在してしまう。そこでt1、t2を自由に設定できるようにしプログラムによって切り換えてもよい。

【0170】以上の機能によりユーザーに分かりやすいシステムを提供できる。

【0171】

10 【発明の効果】以上述べたように本発明の情報処理装置は、連続動作を続けると発熱によって熱暴走してしまうくらいの処理能力をもったCPUを、その処理能力を損なう事なくまた発熱による熱暴走を起こすことのないように温度制御することによって使用可能にする。結果的に、本来大型の放熱板を有するCPUを放熱板を取ることによって携帯型のノートパソコンに装着することができる。

【0172】また特に携帯型のノートパソコン等において、グレードアップされるCPUに応じて最適な発熱コントロールを行うことを可能とする。

20 【図面の簡単な説明】

【図1】 本発明のブロック図。

【図2】 CPUの表面にセンサーを付ける方法の実施例を示す図。

【図3】 本発明の情報処理装置のブロック図。

【図4】 CPUモジュールの実装形体図。

【図5】 ノートパソコンの基板の外形図。

【図6】 本発明のフローチャート。

【図7】 PGA形状のCPUの外形図。

30 【図8】 CPUをグレードアップする場合の一例を示す図。

【図9】 CPUモジュールの簡略化した回路図。

【図10】 CPUのCLKタイプ別の内部ブロック図。

【図11】 CPUのCLKタイプ別の内部ブロック図。

【図12】 CPUのCLKタイプ別の内部ブロック図。

【図13】 CPUの表面温度上昇カーブを示す図。

【図14】 CPUの表面温度上昇カーブを示す図。

40 【図15】 本発明でのCPUの表面温度上昇カーブを示す図。

【図16】 環境温度の違いによる温度上昇カーブを示す図。

【図17】 環境温度の違いによる温度上昇カーブを示す図。

【図18】 CPUの違いによる温度上昇カーブを示す図。

【図19】 CPUの違いによる温度上昇カーブを示す図。

50 【図20】 I/Oポートを示す図。

43

- 【図 2 1】 自動温度制御の為の I/Oポートを示す図。  
 【図 2 2】 ダイナミックパワーセーブの動作波形を示す図。  
 【図 2 3】 電圧可変型ダイナミックパワーセーブの動作波形を示す図。  
 【図 2 4】 自動温度制御時の周波数切り換え波形を示す図。  
 【図 2 5】 温度によって最大 CLK 周波数を設定する図。  
 【図 2 6】 実際の温度上昇カーブを示す図。  
 【図 2 7】 CPUモジュールの実装形体図。  
 【図 2 8】 温度計測部回路図。  
 【図 2 9】 LED制御の回路図。  
 【図 3 0】 実施例 2 の温度上昇カーブ。  
 【図 3 1】 実際の温度上昇カーブ。  
 【図 3 2】 STPCLKN のタイミング波形。

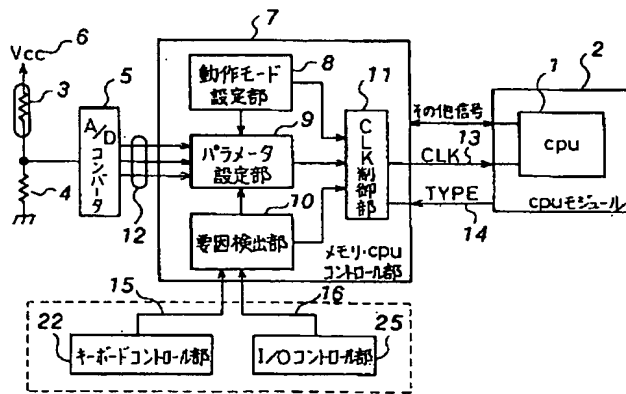
44

- 【図 3 3】 STPCLKN の波形。  
 【図 3 4】 STPCLKN の波形。  
 【図 3 5】 STPCLKN の波形。  
 【図 3 6】 STPCLKN の波形。  
 【図 3 7】 実施例 3 の温度上昇カーブ。

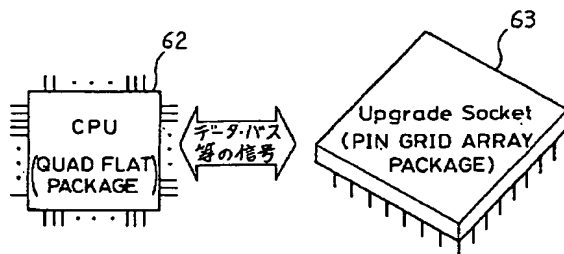
## 【符号の説明】

- 1・・・CPU  
 2・・・CPUモジュール  
 3・・・サーミスタ  
 4・・・抵抗  
 5・・・A/Dコンバータ  
 6・・・電源電圧 Vcc  
 7・・・メモリ・CPUコントロール部  
 8・・・動作モード設定部  
 9・・・パラメータ設定部  
 10・・・パラメータ設定部  
 10・・・要因検出部

【図 1】



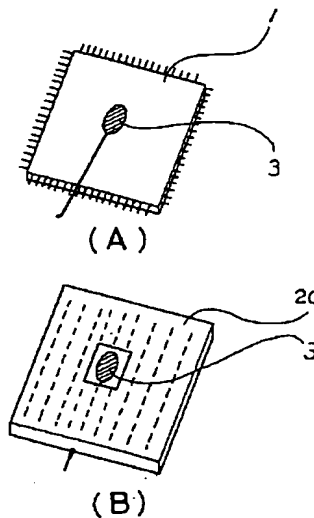
【図 8】



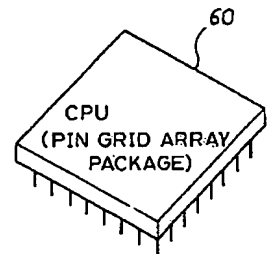
【図 3 3】



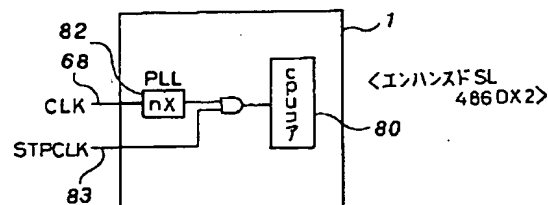
【図 2】



【図 7】

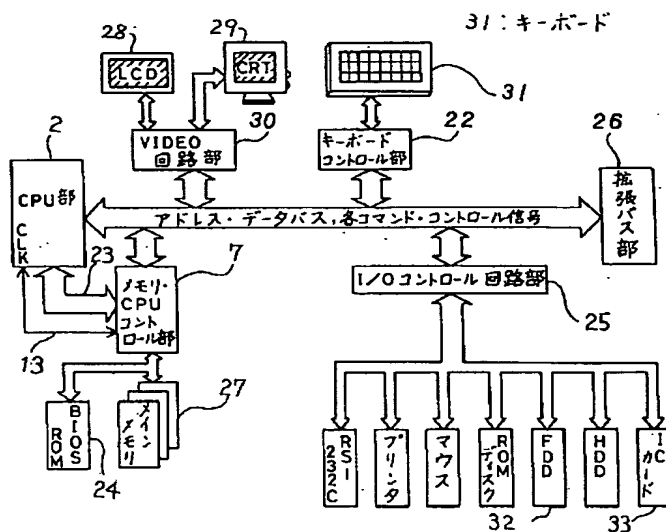


【図 1 1】

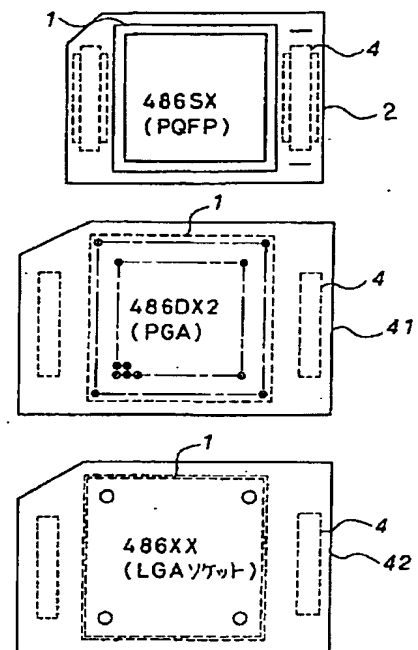




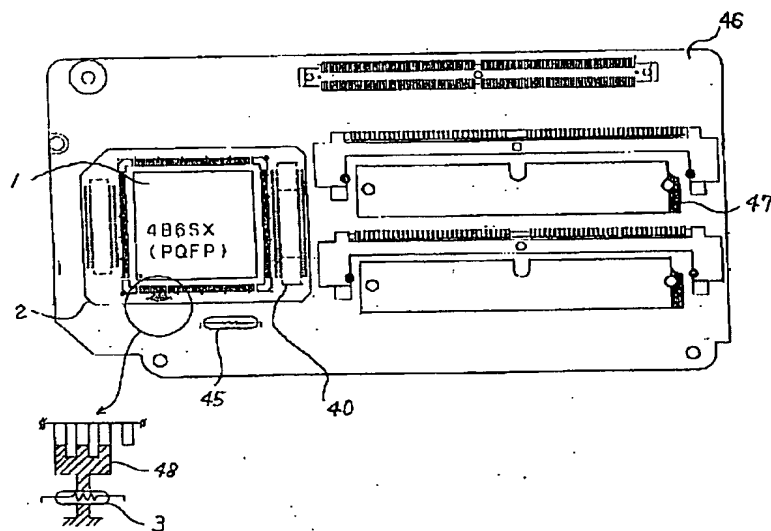
【図 3】



【図 4】

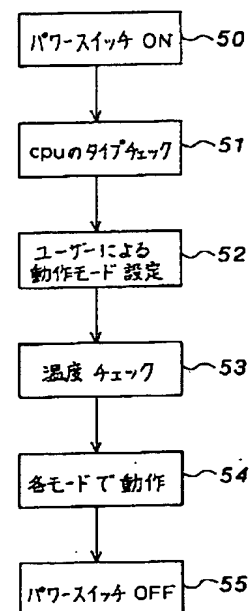


【図 5】



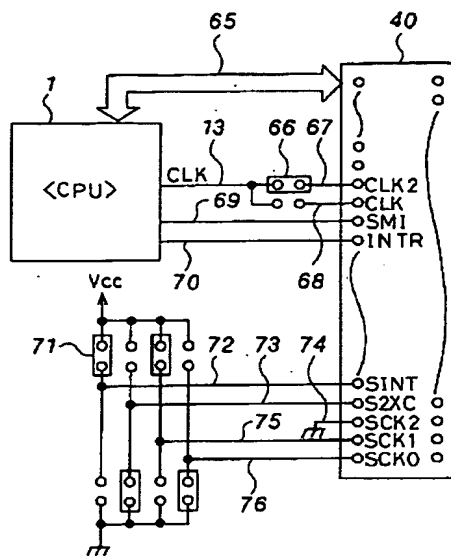
【図 2 5】

【図 6】

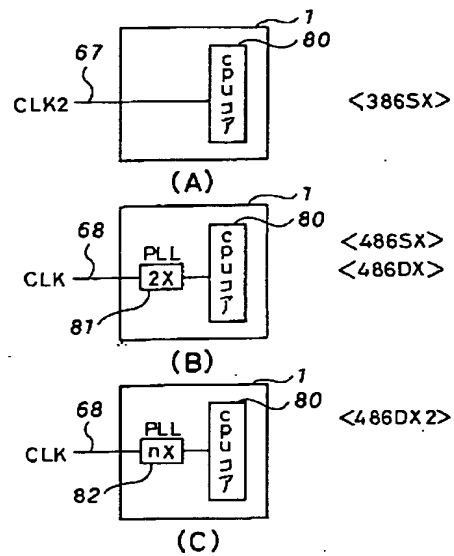


CPU温度 (ケース内温度)	~60°C	65°C	70°C	75°C	80°C~
最大 CLK 周波数	50MHz	45MHz	40MHz	33MHz	25MHz

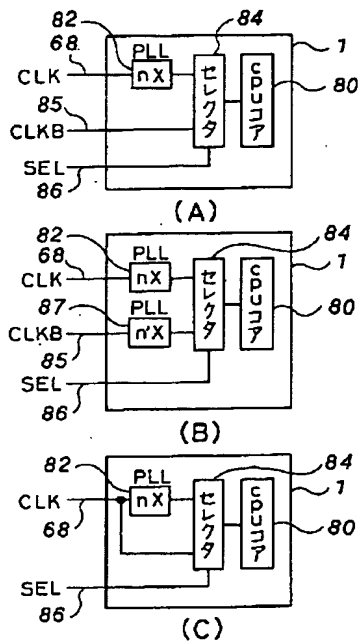
【図 9】



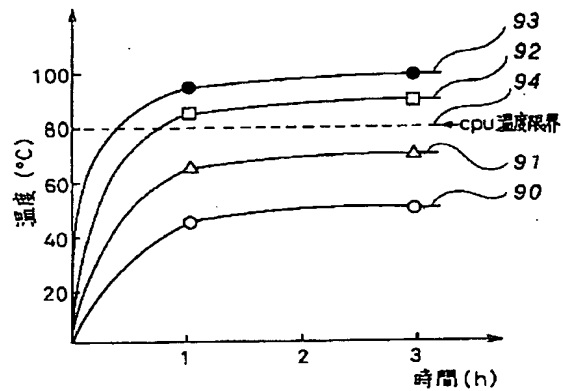
【図 10】



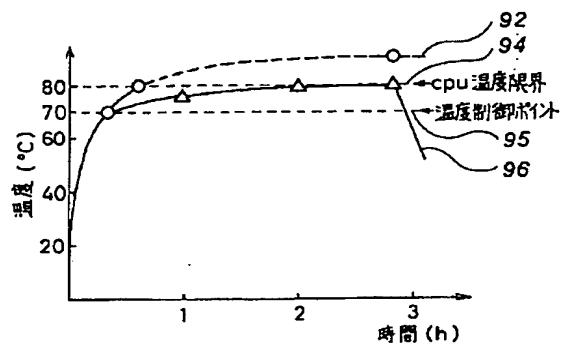
【図 12】



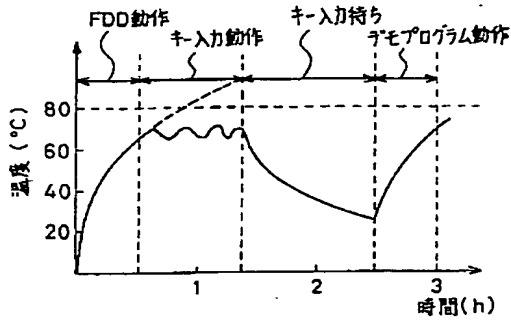
【図 13】



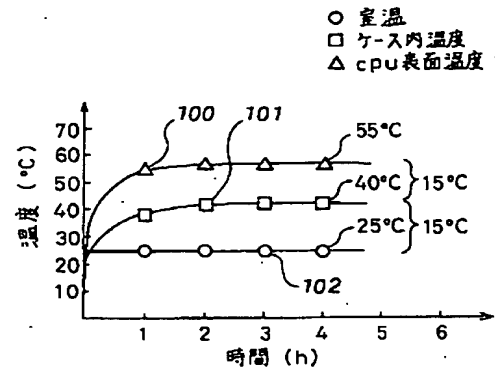
【図 14】



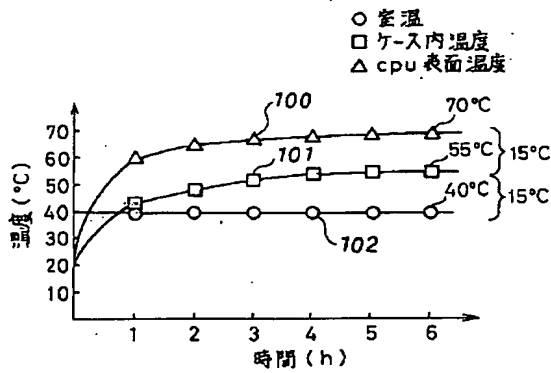
【図 15】



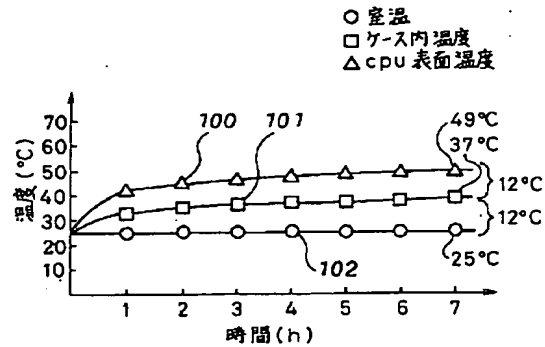
【図 16】



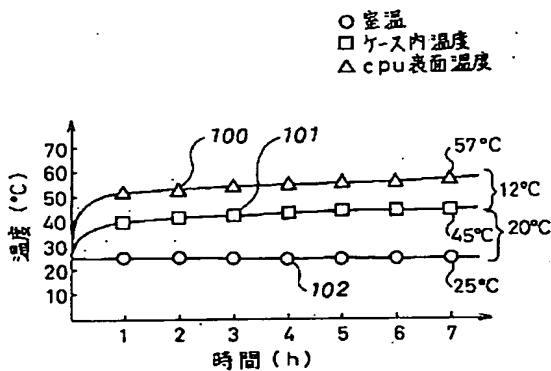
【図 17】



【図 18】



【図 19】



【図 21】

自動温度制御ポート

	bit4	bit3	bit2	bit1	bit0
	キーボード	マウス	FDD	HDD	RS-232C
(200h)	5S	5S	200ms	300ms	1ms
(210h)	10S	15S	400ms	500ms	10ms

		bit2	bit1	bit0
(220h)	FDD	fmax	33	25
(230h)	FDD	fmin	5	2
(240h)	RS-232C	fmax	33	25
(250h)	RS-232C	fmin	5	2
(260h)	K,B	fmax	50	33
(270h)	マウス HDD	fmin	5	2

(単位 MHz)

【図20】

(A) 電圧可変型ダイナミック温度制御ポート (101h)



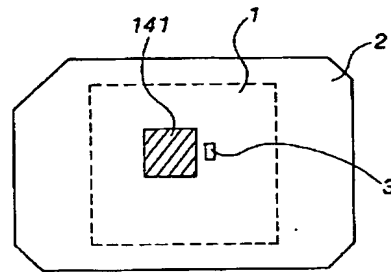
(B) ダイナミックパワーセーブコントロールポート

	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
(110h)	キーボード	FDD	HDD	マウス	RS-232C	VIDEOメモリ	メインメモリ	強制シャット	EX3	EX2	EX1	EX0
(120h)	100ms	200ms	300ms	10ms	1ms	1ms	1ms	1ms	1ms	1ms	1ms	1ms
(130h)	300ms	400ms	500ms	20ms	10ms	10ms	10ms	10ms	10ms	10ms	10ms	10ms

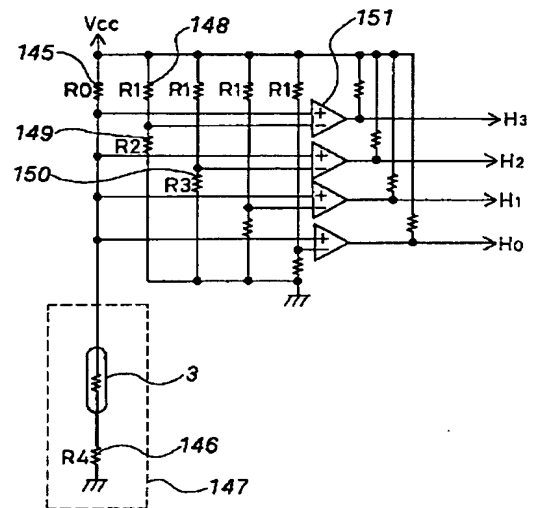
  

	bit12	bit11	bit10		
(140h)	FDD	fmax	33	25	16
(150h)		fmin	5	2	1
(160h)	RS-232C	fmax	33	25	16
(170h)		fmin	5	2	1
(180h)	その他	fmax	50	33	25
(190h)		fmin	5	2	1

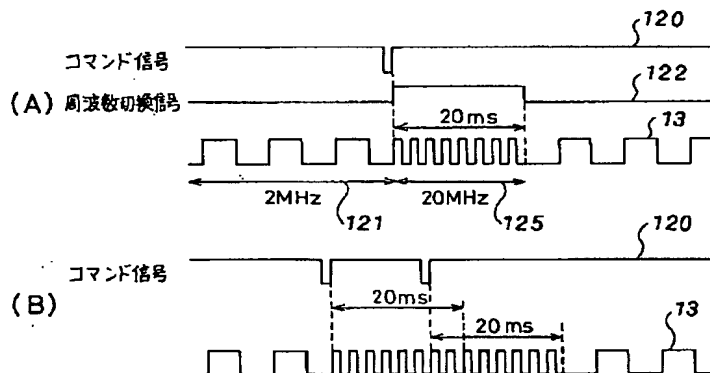
【図27】



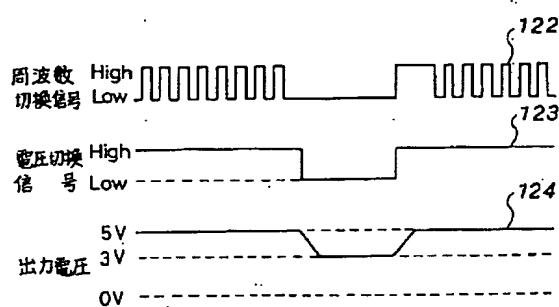
【図28】



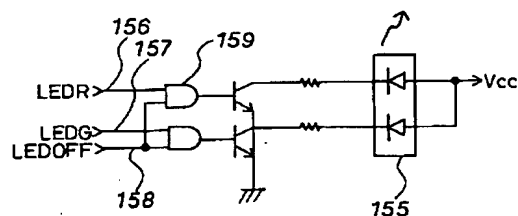
【図22】



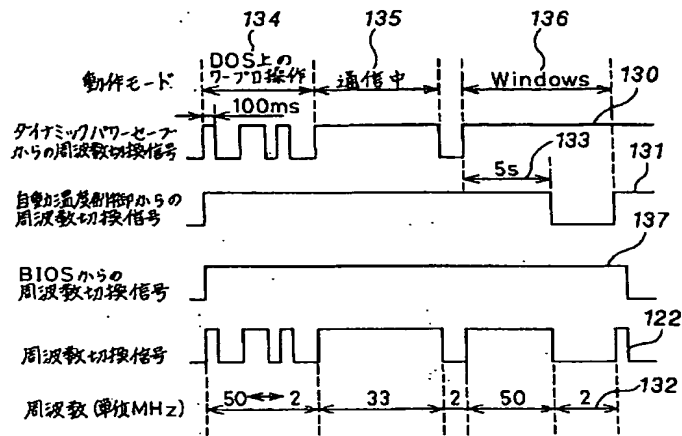
【図23】



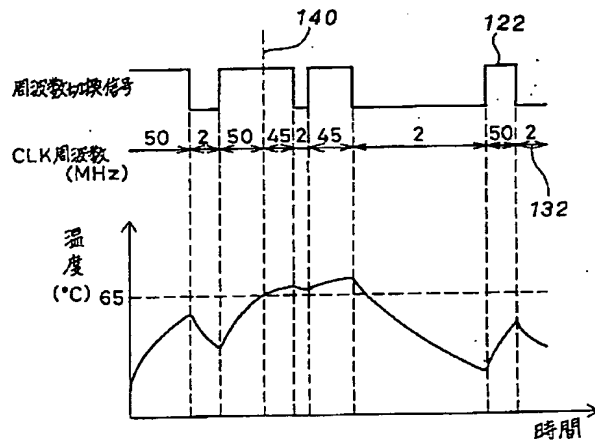
【図29】



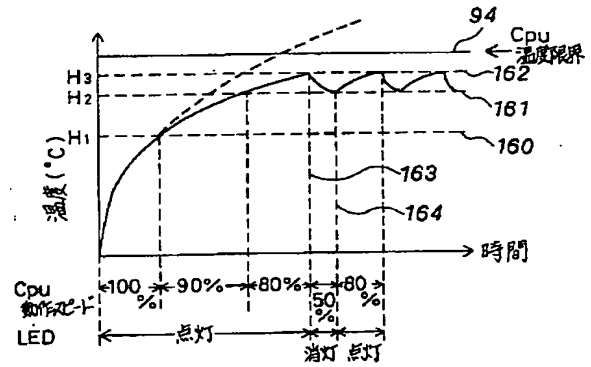
【図 2 4】



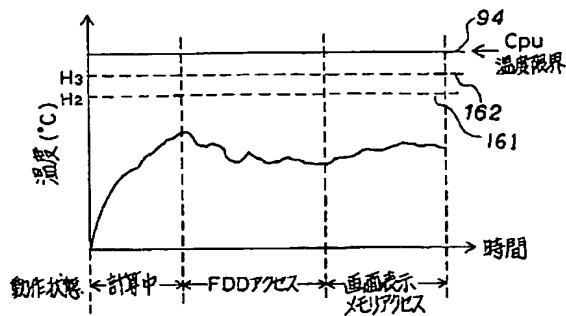
【図 2 6】



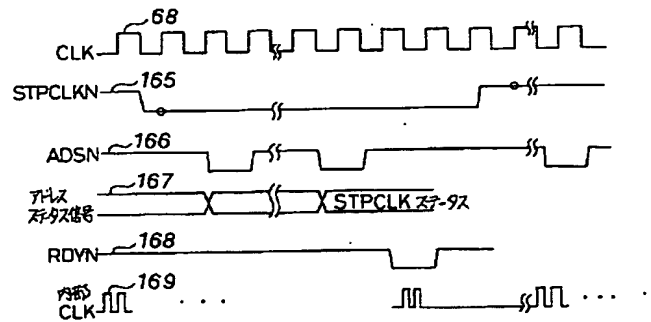
【図 3 0】



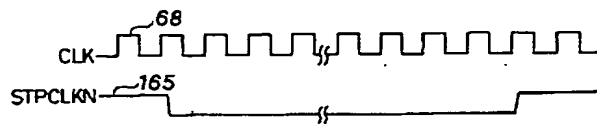
【図 3 1】



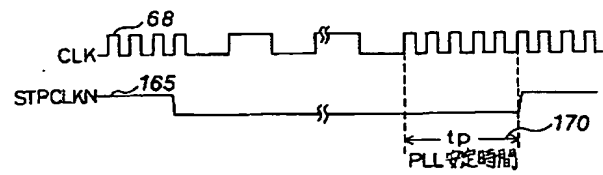
【図 3 2】



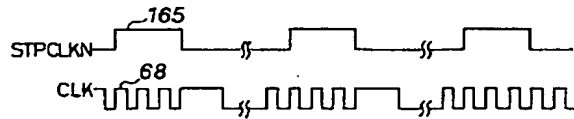
【図 3 4】



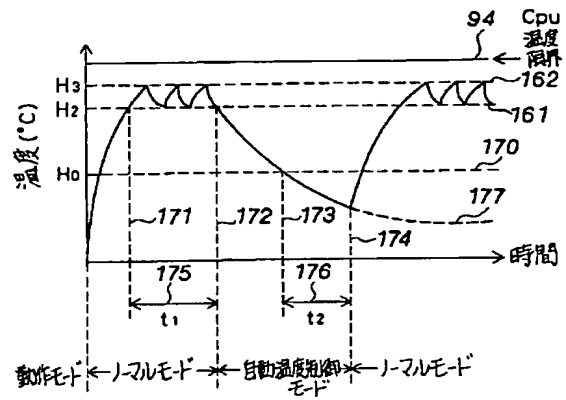
【図 3 5】



【図 3 6】



【図 3 7】



フロントページの続き

(51) Int. Cl.<sup>6</sup>

G 0 6 F 1/08

識別記号

庁内整理番号

F I

技術表示箇所

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☒ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☒ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**